Technical Specification Group Services and System Aspects
Meeting #14, Kyoto, Japan, 17-20 December 2001

TSGS#14(01)0703

Source: **TSG-SA WG4**

Title: CRs to TS 26.234 Corrections and "Implementation guidelines for RTSP and RTP" (Release 4)

Document for: **Approval**

Agenda Item: **7.4.3**

The following CRs, agreed at the TSG-SA WG4 meeting #19, are presented to TSG SA #14 for approval.

Spec	CR	Rev	Phase	Subject	Cat	Vers	WG	Meeting	S4 doc
26.234	007		REL-4	Correction of SDP Usage	F	4.1.0	S4	TSG-SA WG4#19	S4-010638 (R2)
26.234	008	1	REL-4	Implementation guidelines for RTSP and RTP	F	4.1.0	S4	TSG-SA WG4#19	S4-010657
26.234	009		REL-4	Correction to media type decoder support in the PSS client	F	4.1.0	S4	TSG-SA WG4#19	S4-010658
26.234	010		REL-4	Amendments to file format support for 26.234 release 4	F	4.1.0	S4	TSG-SA WG4#19	S4-010589 (R)

3GPP TSG-SA4 Meeting
Tokyo, Japan, 3-7 Dec 2001

Tdoc S4-010589R

CR-Form-v4

CHANGE REQUEST

⌘ **26.234** **CR** **010** ⌘ ev **-** ⌘ Current version: **4.1.0** ⌘

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

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Amendments to file format support
Source:	⌘	TSG SA WG4
Work item code:	⌘	PSTREAM
Date:	⌘	17 Dec 2001
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	Alignment with ISO/IEC, clearer identification of files
Summary of change:	⌘	Files now have their own MIME type, extension, and intrinsic signature. Data specific to 3GPP codecs is in atoms (boxes), following ISO practice.
Consequences if not approved:	⌘	Mis-alignment with ISO/IEC re-write, difficulty of distinguishing 3GPP files from true MP4 files.

Clauses affected:	⌘	2, Annex D
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications
	⌘	
Other comments:	⌘	

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

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

2 References

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

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

[1] (void)

[2] 3GPP TS 26.233: "End-to-end transparent streaming service; General description".

[3] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[4] IETF RFC 1738: "Uniform Resource Locators (URL)", Berners-Lee, Masinter & McCahill, December 1994.

[5] IETF RFC 2326: "Real Time Streaming Protocol (RTSP)", Schulzrinne H., Rao A. and Lanphier R., April 1998.

[6] IETF RFC 2327: "SDP: Session Description Protocol", Handley M. and Jacobson V., April 1998.

[7] IETF STD 0006: "User Datagram Protocol", Postel J., August 1980.

[8] IETF STD 0007: "Transmission Control Protocol", Postel J., September 1981.

[9] IETF RFC 1889: "RTP: A Transport Protocol for Real-Time Applications", Schulzrinne H. et al., January 1996.

[10] IETF RFC 1890: "RTP Profile for Audio and Video Conferences with Minimal Control", Schulzrinne H. et al., January 1996.

[11] 3GPP TS 26.235: "Packet Switched Conversational Multimedia Applications; Default Codecs; Annex ~~D~~B: AMR and AMR-WB RTP payload and MIME type registration ~~RTP payload format for AMR~~".

[12] ~~3GPP TS 26.235: "Packet switched conversational multimedia applications; Default codecs; Annex B: AMR-WB RTP payload and MIME type registration".~~(void)

[13] IETF RFC 3016: "RTP Payload Format for MPEG-4 Audio/Visual Streams", Kikuchi Y. et al., November 2000.

[14] IETF RFC 2429: "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", Bormann C. et al., October 1998.

[15] IETF RFC 2046: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", N. Freed, N. Borenstein, November 1996.

[16] IETF RFC 3023: "XML Media Types", Murata, M., St.Laurent, S., Kohn, D., January 2001.

[17] IETF RFC 2616: "Hypertext Transfer Protocol - HTTP/1.1", Fielding R. et al., June 1999.

[18] 3GPP TS 26.071: "Mandatory Speech Codec speech processing functions; AMR Speech Codec; General description".

[19] 3GPP TS 26.101: "Mandatory Speech Codec speech processing functions; AMR Speech Codec; Frame Structure".

[20] 3GPP TS 26.171: "AMR speech codec, wideband; General description".

[21] ISO/IEC 14496-3 (1999): "Information technology - Coding of audio-visual objects - Part 3: Audio".

[22] ITU-T Recommendation H.263: "Video coding for low bit rate communication".

[23] ITU-T Recommendation H.263 (annex X): "Annex X, Profiles and levels definition".

[24] ISO/IEC 14496-2 (1999): "Information technology - Coding of audio-visual objects - Part 2: Visual".

[25] ISO/IEC 14496-2:1999/FDAM4, ISO/IEC JTC1/SC 29/WG11 N3904, Pisa, January, 2001

[26] ITU-T Recommendation T.81 (1991) | ISO/IEC 10918-1 (1992): "Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines.

[27] "JPEG File Interchange Format", Version 1.02, September 1, 1992.

[28] W3C Recommendation: "XHTML Basic", http://www.w3.org/TR/2000/REC-xhtml-basic-20001219, December 2000

[29] ISO/IEC 10646-1 (2000): "Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane".

[30] The Unicode Consortium: "The Unicode Standard", Version 3.0 Reading, MA, Addison-Wesley Developers Press, 2000, ISBN 0-201-61633-5.

[31] W3C Recommendation: "Synchronized Multimedia Integration Language (SMIL 2.0)", http://www.w3.org/TR/2001/REC-smil20-20010807/, August 2001.

[32] CompuServe Incorporated: "GIF Graphics Interchange Format: A Standard defining a mechanism for the storage and transmission of raster-based graphics information", Columbus, OH, USA, 1987.

[33] CompuServe Incorporated: "Graphics Interchange Format: Version 89a", Columbus, OH, USA, 1990.

[34] ISO/IEC 14496-1 ~~(2000)~~:2001: "Information technology - Coding of audio-visual objects - Part 1: Systems".

[35] 3GPP TS 23.140: "Multimedia Messaging Service (MMS), Functional description stage 2/3".

[36] ISO/IEC 15444-1 (2000): "Information technology - JPEG 2000 image coding system: Core coding system; Annex I: The JPEG 2000 file format".

[37] 3GPP TS 26.201: "AMR Wideband Speech Codec; Frame Structure".

Annex D (normative): Support for non-ISO code streams in MP4 files

D.1 General

The purpose of this annex is to define the necessary structure for integration of the H.263 ~~and~~, AMR and AMR-WB media specific information in an MP4 file. Clauses D.2 to D.4 give some background information about the Sample Description atom, VisualSampleEntry atom and the AudioSampleEntry atom in the MPEG-4 file format. Then, the definitions of the SampleEntry atoms for AMR, AMR-WB and H.263 are given in clauses D.5 to D.8.

AMR and AMR-WB data is stored in the stream according to clause ~~8~~ B.5.2 of [11], without the AMR magic numbers.

D.2 Sample Description atom

In an MP4 file, Sample Description Atom gives detailed information about the coding type used, and any initialisation information needed for that coding. The Sample Description Atom can be found in the MP4 Atom Structure Hierarchy shown in figure D.1.

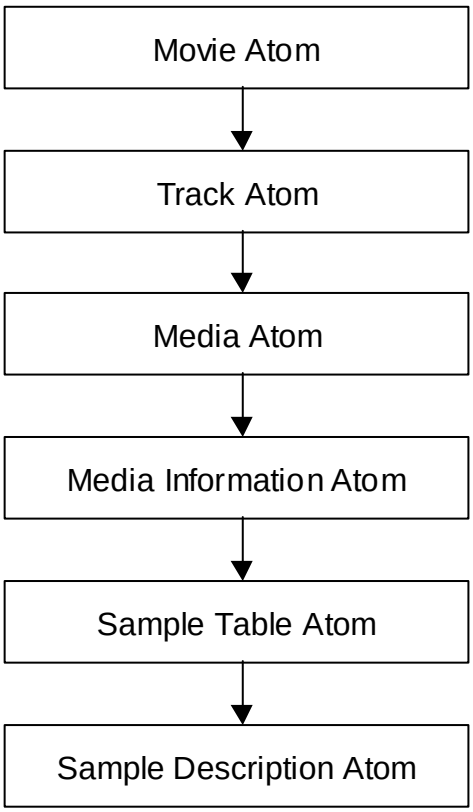

Figure D.1: MP4 Atom Structure Hierarchy

The Sample Description Atom can have one or more SampleDescriptionEntry fields. Valid Sample Description Entry atoms already defined for MP4 are AudioSampleEntry, ~~VideoSampleEntry~~VisualSampleEntry, HintSampleEntry and MPEGSampleEntry Atoms. The Sample DescriptionEntry Atoms for AMR and AMR-WB shall be AMRSampleEntry, and for H.263 shall be ~~AMRSampleEntry and~~ H263SampleEntry, respectively.

The format of SampleDescriptionEntry and its fields are explained as follows:

SampleDescriptionEntry ::= VisualSampleEntry |

AudioSampleEntry |

HintSampleEntry |

MpegSampleEntry

H263SampleEntry |

AMRSampleEntry

Table D.1: SampleDescriptionEntry fields

Field	Type	Details	Value
VisualSampleEntry		Entry type for visual samples defined in the MPEG-4 specification.	
AudioSampleEntry		Entry type for audio samples defined in the MPEG-4 specification.	
HintSampleEntry		Entry type for hint track samples defined in the MPEG-4 specification.	
MpegSampleEntry		Entry type for MPEG related stream samples defined in the MPEG-4 specification.	
H263SampleEntry		Entry type for H.263 visual samples defined in clause D.6 of the present document.	
AMRSampleEntry		Entry type for AMR and AMR-WB speech samples defined in clause D.5 of the present document.	

From the above 5 atoms, only the VisualSampleEntry, AudioSampleEntry, H263SampleEntry and AMRSampleEntry atoms are taken into consideration, since MPEG specific streams and hint tracks are out of the scope of the present document.

D.3 VisualSampleEntry atom

The VisualSampleEntry Atom is defined as follows:

VisualSampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_16

~~Reserved_4~~Width

Height

Reserved_4

Reserved_4

Reserved_4

Reserved_2

Reserved_32

Reserved_2

Reserved_2

ESDAtom

Table D.2: VisualSampleEntry fields

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		'mp4v'
Reserved_6	Unsigned int(8)[6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference Atoms.	
Reserver_16	Const unsigned int(32)[4]		0
~~Reserved_4~~Width	~~Const~~ uUnsigned int(~~32~~16)	Maximum width, in pixels of the stream	~~0x014000f0~~
Height	Unsigned int(16)	Maximum height, in pixels of the stream	
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0
Reserved_2	Const unsigned int(16)		1
Reserved_32	Const unsigned int(8)[32]		0
Reserved_2	Const unsigned int(16)		24
Reserved_2	Const int(16)		-1
ESDAtom		~~Elementary~~ Atom containing an elementary stream descriptor for this stream.	

The stream type specific information is in the ESDAtom structure, which will be explained later. This version of the VisualSampleEntry, with explicit width and height, shall be used for MPEG-4 video streams conformant to this specification.

NOTE: width and height parameters together may be used to allocate the necessary memory in the playback device without need to analyse the video stream.

D.4 AudioSampleEntry atom

AudioSampleEntryAtom is defined as follows:

AudioSampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_8

Reserved_2

Reserved_2

Reserved_4

TimeScale

Reserved_2

ESDAtom

Table D.3: AudioSampleEntry fields

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		'mp4a'
Reserved_6	Unsigned int(8)[6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference Atoms.	
Reserved_8	Const unsigned int(32)[2]		0
Reserved_2	Const unsigned int(16)		2
Reserved_2	Const unsigned int(16)		16
Reserved_4	Const unsigned int(32)		0
TimeScale	Unsigned int(16)	Copied from track	
Reserved_2	Const unsigned int(16)		0
ESDAtom		~~Elementary~~ Atom containing an elementary stream descriptor for this stream.	

The stream type specific information is in the ESDAtom structure, which will be explained later.

D.5 AMRSampleEntry atom

For narrow-band AMR, t~~T~~he atom type of the AMRSampleEntry Atom shall be 'samr'. For AMR wide-band (AMR-WB), the atom type of the AMRSampleEntry Atom shall be 'sawb'.

The AMRSampleEntry Atom is defined as follows:

AMRSampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_8

Reserved_2

Reserved_2

Reserved_4

TimeScale

Reserved_2

~~**DecoderSpecificInfo**~~**AMRSpecificAtom**

Table D.4: AMRSampleEntry fields

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		'samr' or 'sawb'
Reserved_6	Unsigned int(8)[6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference Atoms.	
Reserved_8	Const unsigned int(32)[2]		0
Reserved_2	Const unsigned int(16)		2
Reserved_2	Const unsigned int(16)		16
Reserved_4	Const unsigned int(32)		0
TimeScale	Unsigned int(16)	Copied from media header atom of this media	
Reserved_2	Const unsigned int(16)		0
~~**DecoderSpecificInfo**~~**AMRSpecificAtom**		Information specific to the decoder.	

If one compares the AudioSampleEntry Atom - AMRSampleEntry Atom the main difference is in the replacement of the ESDAtom, which is specific to MPEG-4 systems, with an atom suitable for

AMR and AMR-WB. The **AMRSpecificAtom**~~DecoderSpecificInfo~~ field structure ~~for AMR~~ is described in clause D.7.

D.6 H263SampleEntry atom

The atom type of the H263SampleEntry Atom shall be 's263'.

The H263SampleEntry ~~AMRSampleEntry~~ Atom is defined as follows:

H263SampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_16

~~Reserved_4~~Width

Height

Reserved_4

Reserved_4

Reserved_4

Reserved_2

Reserved_32

Reserved_2

Reserved_2

~~**DecoderSpecificInfo**~~**H263SpecificAtom**

Table D.5: H263SampleEntry fields

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		's263'
Reserved_6	Unsigned int(8)[6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference Atoms.	
Reserver_16	Const unsigned int(32)[4]		0
~~Reserved_4~~Width	~~Const u~~Unsigned int(32)	Maximum width, in pixels of the stream	~~0x014000f0~~
Height	Unsigned int(16)	Maximum height, in pixels of the stream	
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0
Reserved_2	Const unsigned int(16)		1
Reserved_32	Const unsigned int(8)[32]		0
Reserved_2	Const unsigned int(16)		24
Reserved_2	Const int(16)		-1
~~DecoderSpecificInfo~~**H263SpecificAtom**		Information specific to the H.263 decoder.	

If one compares the VisualSampleEntry – H263SampleEntry Atom the main difference is in the replacement of the ESDAtom, which is specific to MPEG-4 systems, with an atom suitable for H.263. The **H263SpecificAtom**~~DecoderSpecificInfo~~ field structure for H.263 is described in clause D.8.

D.7 ~~DecoderSpecificInfo~~ AMRSpecificAtom field for AMRSampleEntry atom

The ~~DecoderSpecificInfo~~ AMRSpecificAtom fields for AMR and AMR-WB shall be as defined in table D.6. The AMRSpecificAtom ~~DecoderSpecificInfo~~ for the AMRSampleEntry Atom shall always be included if the MP4 file contains AMR or AMR-WB media.

Table D.6: The ~~DecoderSpecificInfo~~ AMRSpecificAtom fields for AMRSampleEntry

Field	Type	Details	Value
~~DecSpecificInfoTag~~AtomHeader.Size	~~Bit(8)~~Unsigned int(32)		~~0x05~~
~~SizeOfDecSpecificInfo~~AtomHeader.Type	Unsigned int(32)		'damr'
DecSpecificInfo	AMRDecSpecStruc	Structure which holds the AMR and AMR-WB Specific information	

~~**DecSpecificInfoTag:** identifies that this is a DecoderSpecificInfo Field. It must be set to 0x05.~~

~~**SizeOfDecSpecificInfo:** defines the size (in Bytes) of the DecSpecificInfo structure following.~~

AtomHeader Size and Type: indicate the size and type of the AMR decoder-specific atom. The type must be 'damr'.

DecSpecificInfo: the structure where the AMR and AMR-WB stream specific information resides.

The AMRDecSpecStruc is defined as follows:

struct **AMRDecSpecStruc**{

Unsigned int (32)	**vendor**
Unsigned int (8)	**decoder_version**
Unsigned int (16)	**mode_set**
Unsigned int (8)	**mode_change_period**
Unsigned int (8)	**frames_per_sample**

}

The definitions of AMRDecSpecStruc members are as follows:

vendor: four character code of the manufacturer of the codec, e.g. 'VXYZ'. The vendor field gives information about the vendor whose codec is used to create the encoded data. It is an informative field which may be used by the decoding end. If a manufacturer already has a four~~4~~ character~~cc~~ code, it is recommended that it uses the same code in this field. Else, it is recommended that the manufacturer creates a four~~4~~ character~~cc~~ code which best addresses the manufacturer's name. It can be safely ignored ~~if it does not provide further information to the decoding end~~.

decoder_version: version of the vendor's decoder which ~~created the AMR~~ can decode the encoded stream in the best (i.e. optimal) way. This field is closely tied to the vendor field. It may give advantage to the vendor which has optimal encoder-decoder version pairs.~~stream being stored, the~~ . The value is set to 0 if decoder version has no importance for the vendor. It can be safely ignored ~~if the vendor field does not provide further information to the decoding end~~.

mode_set: the active codec modes. ~~A value of 0x1F means all modes are possibly present in the AMR stream.~~ Each bit of the mode_set parameter corresponds to one mode. The bit index of the mode is calculated according to the 4 bit FT field of the AMR or AMR-WB frame structure. ~~The mapping of existing AMR modes to FT is given in table 1.a in [19].~~ The mode_set bit structure is as follows: (B15xxxxxxB8B7xxxxxxB0) where B0 (Least Significant Bit) corresponds to Mode 0, and B8 corresponds to Mode 8.

The mapping of existing AMR modes to FT is given in table 1.a in [19]. A value of 0x81FF means all modes and comfort noise frames are possibly present in an AMR stream.

The mapping of existing AMR-WB modes to FT is given in Table 1.a in TS 26.201 [37]. A value of 0x83FF means all modes and comfort noise frames are possibly present in an AMR-WB stream.

As an example, if mode_set = 0000000110010101b, only ~~AMR~~ Modes 0, 2, 4, 7 and 8 are present in the ~~AMR~~ stream.

mode_change_period: defines a number N, which restricts the mode changes only at a multiple of N frames. If no restriction is applied, this value should be set to 0. If mode_change_period is not 0, the following restrictions apply to it according to the frames_per_sample field:

if (mode_change_period < frames_per_sample)

frames_per_sample = k x (mode_change_period)

else if (mode_change_period > frames_per_sample)

mode_change_period = k x (frames_per_sample)

where k : integer [2, …]

If mode_change_period is equal to frames_per_sample, then ~~AMR~~ the mode is the same for all frames inside one sample.

frames_per_sample: defines the number of frames to be considered as 'one sample' inside the MP4 file. This number should be greater than 0. A value of 1 means each frame is treated as one sample. A value of 10 means that 10 ~~AMR~~ frames (of duration 20 msec each) are put together and treated as one sample. It must be noted that, in this case, one sample duration is 20 (msec/frame) x 10 (frame) = 200 msec. For the last sample of the ~~AMR~~ stream, the number of frames can be smaller than frames_per_sample, if the number of remaining frames is smaller than frames_per_sample.

NOTE: The "hinter", for the creation of the hint tracks, can use the information given by the AMRDecSpecStruc members.

D.8 ~~DecoderSpecificInfo~~ H263SpecificAtom field for H263SampleEntry atom

The ~~DecoderSpecificInfo~~ H263SpecificAtom fields for H. 263 shall be as defined in table D.7. The ~~DecoderSpecificInfo~~ H263SpecificAtom for the H263SampleEntry Atom shall always be included if the MP4 file contains H.263 media.

The H263SpecificAtom ~~DecoderSpecificInfo~~ for H263 is composed of the following fields.

Table D.7: The ~~DecoderSpecificInfo~~ H263SpecificAtom fields H263SampleEntry

Field	Type	Details	Value
~~DecSpecificInfoTag~~AtomHeader.Size	~~Bit(8)~~Unsigned int(32)		~~0x05~~
~~SizeOfDecSpecificInfo~~AtomHeader.Type	Unsigned int(32)		'd263'
DecSpecificInfo	H263DecSpecStruc	Structure which holds the H.263 Specific information	

~~**DecSpecificInfoTag:** It identifies that this is a DecoderSpecificInfo field. It shall be set to 0x05.~~

~~**SizeOfDecSpecificInfo:** It defines the size (in Bytes) of the DecSpecificInfo structure following.~~

AtomHeader Size and Type: indicate the size and type of the H.263 decoder-specific atom. The type must be 'd263'.

DecSpecificInfo: This is the structure where the H263 stream specific information resides.

H263DecSpecStruc is defined as follows:

struct **H263DecSpecStruc**{

Unsigned int (32)	**vendor**
Unsigned int (8)	**decoder_version**
Unsigned int (8)	**H263_Level**
Unsigned int (8)	**H263_Profile**
~~Unsigned int (16)~~	~~**max_width**~~
~~Unsigned int (16)~~	~~**max_height**~~

}

The definitions of H263DecSpecStruc members are as follows:

vendor: four character code of the manufacturer of the codec, e.g. 'VXYZ'. The vendor field gives information about the vendor whose codec is used to create the encoded data. It is an informative field which may be used by the decoding end. If a manufacturer already has a four character code, it is recommended that it uses the same code in this field. Else, it is recommended that the manufacturer creates a four character code which best addresses the manufacturer's name. It can be safely ignored.

decoder_version: version of the vendor's decoder which can decode the encoded stream in the best (i.e. optimal) way. This field is closely tied to the vendor field. It may give advantage to the vendor which has optimal encoder-decoder version pairs. . The value is set to 0 if decoder version has no importance for the vendor. It can be safely ignored.

~~**vendor:** Four character code of the manufacturer of the codec, e.g. 'VXYZ'.~~

~~**decoder_version:** Version of the decoder which created the H263 stream being stored. This value is set to 0 if version has no importance.~~

H263_Level and H263_Profile: These two parameters define which H263 profile and level is used. These parameters are based on the MIME media type video/H263-2000. The profile and level specifications can be found in [23].

EXAMPLE 1: H.263 Baseline = {H263_Level = 10, H263_Profile = 0}

EXAMPLE 2: H.263 Profile 3 @ Level 10 = {H263_Level = 10 , H263_Profile = 3}

~~**max_width:** The maximum width of encoded image.~~

~~**max_height:** The maximum height of encoded image.~~

NOTE ~~1~~: ~~max_width and max_height parameters together may be used to allocate the necessary memory in the playback device without need to analyse the H.263 stream.~~

~~NOTE 2:~~ The "hinter", for the creation of the hint tracks, can use the information given by the H263DecSpecStruc members.

D.9 File Identification

3GPP multimedia files can be identified using several mechanisms. When stored in traditional computer file systems, these files should be given the file extension ".3gp" (readers should allow mixed case for the alphabetic characters). The MIME types "video/3gpp" (for video or audio/video content) and "audio/3gpp" (for audio content) are expected to be registered and used.

A file-type atom, as defined in the JPEG 2000 specification [36] shall be present in conforming files. The file type box 'ftyp' shall occur before any variable-length box (e.g. movie, free space, media data). Only a fixed-size box such as a file signature, if required, may precede it.

The brand identifier for this specification is '3gp4'. This brand identifier must occur in the compatible brands list, and may also be the primary brand. Readers should check the compatible brands list for this identifier, and not rely on the file having a primary brand of '3gp4', for maximum compatibility. Files may be compatible with more than one brand, and have a 'best use' other than this specification, yet still be compatible with this specification.

Table D.8: The File-Type atom

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		'ftyp'
Brand	Unsigned int(32)	The major or 'best use' of this file	
MinorVersion	Unsigned int(32)		
CompatibleBrands	Unsigned int(32)	A list of brands, to end of the atom	

Brand: Identifies the 'best use' of this file. The brand should match the file extension. For files with extension '.3gp' and conforming to this specification, the brand shall be '3gp4'.

MinorVersion: This identifies the minor version of the brand. For files with brand '3gp4', and conforming to release 4.x.y, this field takes the value x*256 + y.

CompatibleBrands: a list of brand identifiers (to the end of the atom). '3gp4' shall be a member of this list.

3GPP TSG-SA WG4 Meeting #19
December 3rd – 7th ,2001; Tokyo, Japan

Tdoc S4-010638R2

CR-Form-v4

CHANGE REQUEST

⌘ **26.234** **CR** **007** ⌘ ev **-** ⌘ Current version: **4.1.0** ⌘

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

Proposed change affects: ⌘ (U)SIM ☐ ME/UE **X** Radio Access Network ☐ Core Network ☐

Title:	⌘	CR TS 26.234-007 Correction of SDP usage (Release 4)
Source:	⌘	TSG SA WG4
Work item code:	⌘	PSTREAM
Date:	⌘	17 Dec. 2001
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

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

Reason for change:	⌘	In RFC2327 , a mandatory order of SDP fields are defined, but the examples described in Annex A violate the pre-defined order. In addition, RFC2327 specifies that either 'e' or 'p' field shall be included in the SDP description. But according to the new SDP internet draft (http://www.ietf.org/internet-drafts/draft-ietf-mmusic-sdp-new-03.txt), both of these fields are made optional. A solution for interoperability robustness is needed.
Summary of change:	⌘	The order of SDP fields is mandated and is corrected in the examples. Additionally, note for 'e' and 'p' fields is added in the Table A.1.
Consequences if not approved:	⌘	PSS servers and clients do not conform to RFC2327. Possible interoperability problems may arise because of old and new definitions of 'e' and 'p' fields in RFC 2327 and new SDP draft.

Clauses affected:	⌘	A.1 and A.2
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

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

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

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

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

Annex A (informative): Protocols

A.1 SDP

This clause gives some background information on SDP.

Table A.1 provides an overview of the different SDP fields that can be identified in a SDP file. The order of SDP fields are mandated as specified in RFC 2327 [6].

Table A.1: Overview of fields in SDP

Type	Description		Requirement according to [Error! Bookmark not defined.]	Requirement according to the present document
Session Description				
V	Protocol version		R	R
O	Owner/creator and session identifier		R	R
S	Session Name		R	R
I	Session information		O	O
U	URI of description		O	O
E	Email address		O	O
P	Phone number		O	O
C	Connection Information		R	R
B	Bandwidth information	AS	O	R
One or more Time Descriptions (See below)				
Z	Time zone adjustments		O	O
K	Encryption key		O	O
A	Session attributes	control	O	R
		range	O	R
One or more Media Descriptions (See below)				
Time Description				
T	Time the session is active		R	R
R	Repeat times		O	O
Media Description				
M	Media name and transport address		R	R
I	Media title		O	O
C	Connection information		R	R
B	Bandwidth information	AS	O	R
K	Encryption Key		O	O
A	Attribute Lines	control	O	R
		range	O	R
		fmtp	O	R
		rtpmap	O	R

Note 1: R = Required, O = Optional

Note 2: The "c" type is only required on the session level if not present on the media level.

Note 3: The "c" type is only required on the media level if not present on the session level.

Note 4: According to RFC 2327, either an 'e' or 'p' field must be present in the SDP description. On the other hand, both fields will be made optional in the future release of SDP. So, for the sake of robustness and maximum interoperability, either an 'e' or 'p' field shall be present during the server's SDP file creation, but the client should also be ready to receive SDP content that does not have neither 'e' nor 'p' fields.

The example below shows an SDP file that could be sent to a PSS client to initiate unicast streaming of a H.263 video sequence.

EXAMPLE:
v=0
o=ghost 2890844526 2890842807 IN IP4 192.168.10.10
s=3GPP Unicast SDP Example
i=Example of Unicast SDP file
u=http://www.infoserver.com/ae600
e=ghost@mailserver.com
c=IN IP4 ~~192.168.30.29~~ 0.0.0.0
~~a=range:npt=0-45.678~~
b=AS:128
t=0 0
a=range:npt=0-45.678
m=video 1024 RTP/AVP 96
b=AS:128
a=rtpmap:96 H263-2000/90000
a=fmtp:96 profile=3;level=10
a=control:rtsp;//mediaserver.com/movie
a=recvonly
~~b=AS:128~~

NOTE: The SDP parsers and/or interpreters should be able to accept NULL values in the 'c=' field (e.g. 0.0.0.0 in IPv4 case). This may happen when the media content does not have a fixed destination address. For more details, see Section C.1.7 of [5] and Section 6 of [6].

A.2 RTSP

The example below is intended to give some more understanding of how RTSP and SDP are used within the 3GPP PSS. The example assumes that the streaming client has the RTSP URL to a presentation consisting of an H.263 video sequence and AMR speech. RTSP messages sent from the client to the server are in **bold** and messages from the server to the client in *italic*. In the example the server provides aggregate control of the two streams.

EXAMPLE:

DESCRIBE rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 1

RTSP/1.0 200 OK
CSeq: 1
Content-Type: application/sdp
Content-Length: ~~203~~435

v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video
e=foo@bar.com
c=IN IP4 192.168.30.29
b=AS:77
t=0 0
a=range:npt=0-59.3478
*a=control:**

~~*b=AS:77*~~
~~*t=0 0*~~
m=audio 0 RTP/AVP 97
b=AS:13
a=rtpmap:97 AMR/8000
a=fmtp:97 mode-set=0,2,5,7; ~~*maxframes=1*~~ *maxptime=200*
a=control:streamID=0
~~*b=AS:13*~~
m=video 0 RTP/AVP 98
b=AS:64
a=rtpmap:98 H263-2000/90000
a=fmtp:98 profile=3;level=10
a=control: streamID=1
~~*b=AS:64*~~

SETUP rtsp://mediaserver.com/movie.test/streamID=0 RTSP/1.0
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457

RTSP/1.0 200 OK
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457; server_port=5678-5679
Session: dfhyrio90llk

SETUP rtsp://mediaserver.com/movie.test/streamID=1 RTSP/1.0
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459; server_port=5680-5681
Session: dfhyrio90llk

PLAY rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 4
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 4
Session: dfhyrio90llk
Range: npt=0-
RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0; seq=9900093;rtptime=4470048,
url= rtsp://mediaserver.com/movie.test/streamID=1; seq=1004096;rtptime=1070549

NOTE: Headers can be folded onto multiple lines if the continuation line begins with a space or horizontal tab. For more information, see RFC2616 [**Error! Reference source not found.**].

~~*RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0; seq=9900093;rtptime=4470048,*~~
~~*url= rtsp://mediaserver.com/movie.test/streamID=1; seq=1004096;rtptime=1070549*~~

The user watches the movie for 20 seconds and then decides to fast forward to 10 seconds before the end…

PAUSE rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 5
Session: dfhyrio90llk

PLAY rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 6
Range: npt=50-59.3478
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 5
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 6
Session: dfhyrio90llk
Range: npt=50-59.3478
RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0;
seq=39900043;rtptime=44470648,
url= rtsp://mediaserver.com/movie.test/streamID=1;
seq=31004046;rtptime=41090349

~~*RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0;*~~
~~*seq=39900043;rtptime=44470648, url= rtsp://mediaserver.com/movie.test/streamID=1;*~~
~~*seq=31004046;rtptime=41090349*~~

After the movie is over the client issues a TEARDOWN to end the session…

TEARDOWN rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 7
Session: dfhyrio90llk

RTSP/1.0 200 OK
Cseq: 7
Session: dfhyrio90llk
Connection: close

3GPP TSG-SA4 Meeting #19
Tokyo, Japan, December 3-7, 2001

Tdoc S4-010657

CR-Form-v5

CHANGE REQUEST

⌘ **26.234** **CR** **008** ⌘ **rev** **1** ⌘ Current version: **4.1.0** ⌘

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

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Implementation guidelines for RTSP and RTP
Source:	⌘	TSG SA WG4
Work item code:	⌘	PSTREAM
Date:	⌘	December 17, 2001
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change: ⌘

1) Large RTP packets over the wireless link result either in large delay jitter or in large media losses depending on the radio bearer used. A recommendation for appropriate sized RTP packets is needed for the service to work as intended.

2) RTSP can use either persistent or non-persistent TCP. A recommendation on which one to use is needed for the service to work as intended.

3) The correct usage of sequence numbers and timestamps of RTP is not clear. This may lead to interoperability problems.

4) RTSP should use some “wellness” information to detect aliveness of the link, which is may be lost in a wireless environment. To prevent inter-operability problems a recommendation on how to do this is needed.

Summary of change: ⌘

1) The CR clarifies that RTP packet sizes should be limited in size taking the wireless link into account. Resynchronization information using GOBs for H.263 and video packets for MPEG-4 should be used when a video frames are split into more than one RTP packet.

2) The CR clarifies that PSS recommends the usage of a persistent connection when TCP is used as transport for RTSP messages. This will result in a faster set-up of the PSS session compared to when a non-persistent connection is used.

3) The CR clarifies the correct usage of RTP sequence numbers and timestamp across NPT jumps. RTP sequence numbers and RTP timestamps should be specified continuously and monotonically across the skip of the media.

4) The CR clarifies that the PSS client should send “wellness” information to the PSS server as defined in the RTSP RFC 2326. This will help the server to detect the session aliveness.

Consequences if not approved: ⌘

1) Higher buffering demands in a PSS client due to large delay jitter or decreased media quality due to packet losses.

2) Additional set-up delays for a PSS service, but also potential interoperability

	problems. 3) Possible interoperability problem by misusing of the sequence number and timestamp of RTP. 4) Possible interoperability problems with lost RTSP sessions as a result, but also unnecessary allocation of resources in the PSS server's.

Clauses affected:	⌘	Annex A		
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

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

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

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

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

A.2 RTSP

A.2.1 General

The example below is intended to give some more understanding of how RTSP and SDP are used within the 3GPP PSS. The example assumes that the streaming client has the RTSP URL to a presentation consisting of an H.263 video sequence and AMR speech. RTSP messages sent from the client to the server are in **bold** and messages from the server to the client in *italic*. In the example the server provides aggregate control of the two streams.

EXAMPLE:

DESCRIBE rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 1

RTSP/1.0 200 OK
CSeq: 1
Content-Type: application/sdp
Content-Length: 203

v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video
c=IN IP4 192.168.30.29
a=range:npt=0-59.3478
*a=control:**

b=AS:77
t=0 0
m=audio 0 RTP/AVP 97
a=rtpmap:97 AMR/8000
a=fmtp:97 mode-set=0,2,5,7; maxframes=1
a=control:streamID=0
b=AS:13
m=video 0 RTP/AVP 98
a=rtpmap:98 H263-2000/90000
a=fmtp:98 profile=3;level=10
a=control: streamID=1
b=AS:64

SETUP rtsp://mediaserver.com/movie.test/streamID=0 RTSP/1.0
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457

RTSP/1.0 200 OK
CSeq: 2
Transport: RTP/AVP/UDP;unicast;client_port=3456-3457; server_port=5678-5679
Session: dfhyrio90llk

SETUP rtsp://mediaserver.com/movie.test/streamID=1 RTSP/1.0
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 3
Transport: RTP/AVP/UDP;unicast;client_port=3458-3459; server_port=5680-5681
Session: dfhyrio90llk

PLAY rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 4
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 4
Session: dfhyrio90llk
Range: npt=0-
RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0; seq=9900093;rtptime=4470048, url= rtsp://mediaserver.com/movie.test/streamID=1; seq=1004096;rtptime=1070549

The user watches the movie for 20 seconds and then decides to fast forward to 10 seconds before the end…

PAUSE rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 5
Session: dfhyrio90llk

PLAY rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 6
Range: npt=50-59.3478
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 5
Session: dfhyrio90llk

RTSP/1.0 200 OK
CSeq: 6
Session: dfhyrio90llk
Range: npt=50-59.3478
RTP-Info: url= rtsp://mediaserver.com/movie.test/streamID=0; seq=39900043;rtptime=44470648, url= rtsp://mediaserver.com/movie.test/streamID=1; seq=31004046;rtptime=41090349

After the movie is over the client issues a TEARDOWN to end the session…

TEARDOWN rtsp://mediaserver.com/movie.test RTSP/1.0
CSeq: 7
Session: dfhyrio90llk

RTSP/1.0 200 OK
Cseq: 7
Session: dfhyrio90llk
Connection: close

A.2.2 Implementation guidelines

A.2.2.1 Usage of persistent TCP

Considering the potentially long round-trip-delays in a packet switched streaming service over UMTS it is important to keep the number of messages exchanged between a server and a client low. The number of requests and responses exchanged is one of the factors that will determine how long it takes from the time that a user initiates PSS until the streams starts playing in a client.

RTSP methods are sent over either TCP or UDP for IP. Both client and server must support RTSP over TCP whereas RTSP over UDP is optional. For TCP the connection can be persistent or non-persistent. A persistent connection is used for several RTSP request/response pairs whereas one connection is used per RTSP request/response pair for the non-persistent connection. In the non-persistent case each connection will start with the three-way handshake (SYN, ACK, SYN) before the RTSP request can be sent. This will increase the time for the message to be sent by one round trip delay.

For these reasons it is recommended that 3GPP PSS clients should use a persistent TCP connection, at least for the initial RTSP methods until media starts streaming.

A.2.2.2 Detecting link aliveness

In the wireless environment, connection may be lost due to fading, shadowing, loss of battery power, or turning off the terminal even though the PSS session is active. In order for the server to be able to detect the client's aliveness, the PSS client should send "wellness" information to the PSS server for a defined interval as described in the RFC2326. There are several ways for detecting link aliveness described in the RFC2326, however, the client should be careful about issuing "PLAY method without Range header field" too close to the end of the streams, because it may conflict with pipelined PLAY requests. Below is the list of recommended "wellness" information for the PSS clients and servers in a prioritised order.

1. RTCP
2. OPTIONS method with Session header field

NOTE: Both servers and clients can initiate this OPTIONS method.

A.3 RTP

A.3.1 General

Void.

A.3.2 Implementation guidelines

A.3.2.1 Maximum RTP packet size

The RFC 1889 (RTP) [**Error! Bookmark not defined.**] does not impose a maximum size on RTP packets. However, when RTP packets are sent over the radio link of a 3GPP PSS system there is an advantage in limiting the maximum size of RTP packets.

Two types of bearers can be envisioned for streaming using either acknowledged mode (AM) or unacknowledged mode (UM) RLC. The AM uses retransmissions over the radio link whereas the UM does not. In UM mode large RTP packets are more susceptible to losses over the radio link compared to small RTP packets since the loss of a segment may result in the loss of the whole packet. On the other hand in AM mode large RTP packets will result in larger delay jitter compared to small packets as there is a larger chance that more segments have to be retransmitted.

For these reasons it is recommended that the maximum size of RTP packets should be limited is size taking into account the wireless link. This will decrease the RTP packet loss rate particularly for RLC in UM. For RLC in AM the delay

jitter will be reduced permitting the client to use a smaller receiving buffer. It should also be noted that too small RTP packets could result in too much overhead if IP/UDP/RTP header compression is not applied or unnecessary load at the streaming server.

In the case of transporting video in the payload of RTP packets it may be that a video frame is split into more than one RTP packet in order not to produce too large RTP packets. Then, to be able to decode packets following a lost packet in the same video frame, it is recommended that synchronisation information be inserted at the start of such RTP packets. For H.263 this implies the use of GOBs with non-empty GOB headers and in the case of MPEG-4 video the use of video packets (resynchronisation markers). If the optional Slice Structured mode (Annex K) of H.263 is in use, GOBs are replaced by slices.

A.3.2.2 Sequence number and timestamp in the presence of NPT jump

The description below is intended to give more understanding of how RTP sequence number and timestamp are specified within the 3GPP PSS in the presence of NPT jumps. The jump happens when a client sends a PLAY request to skip media.

The RFC 2326 (RTSP) [**Error! Bookmark not defined.**] specifies that both RTP sequence numbers and RTP timestamps must be continuous and monotonic across jumps of NPT. Thus when a server receives a request for a skip of the media that causes a jump of NPT, it shall specify RTP sequence numbers and RTP timestamps continuously and monotonically across the skip of the media to conform to the RTSP specification. Also, the server may respond with "seq" in the RTP-Info field if this parameter is known at the time of issuing the response.

3GPP TSG-SA4 Meeting #19
Tokyo, Japan, 3-7 December 2001

Tdoc S4-010658

CR-Form-v3

CHANGE REQUEST

⌘ **26.234** **CR** **009** ⌘ rev **-** ⌘ Current version: **4.1.0** ⌘

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

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Correction to media type decoder support in the PSS client
Source:	⌘	TSG SA WG4
Work item code:	⌘	PSTREAM
Date:	⌘	17 Dec 2001
Category:	⌘	**F**
Release:	⌘	Rel-4

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

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

Reason for change:	⌘	The current version of the specification requires the support for AMR-NB, AMR-WB, H.263, MPEG4, GIF and text codecs for the PSS client whereas only the decoders are needed.
Summary of change:	⌘	Section 7 is updated to reflect the fact that encoders are not required in the PSS client.
Consequences if not approved:	⌘	Implementations may include encoders even though they may not be needed.

Clauses affected:	⌘	7.2
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

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

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

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 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 Codecs

7.1 General

For PSS offering a particular media type, media decoders ~~codecs~~ are specified in the following clauses.

7.2 Speech

The AMR decoder ~~codec~~ shall be supported for narrow-band speech [**Error! Bookmark not defined.**]. The AMR wideband speech decoder ~~codec~~ [**Error! Bookmark not defined.**] shall be supported when wideband speech working at 16 kHz sampling frequency is supported.

7.3 Audio

MPEG-4 AAC Low Complexity object type decoder [21] should be supported. The maximum sampling rate to be supported by the decoder is 48 kHz. The channel configurations to be supported are mono (1/0) and stereo (2/0). In addition, the MPEG-4 AAC Long Term Prediction object type decoder may be supported.

7.4 Video

ITU-T Recommendation H.263 [**Error! Reference source not found.**] profile 0 level 10 shall be supported. This is the mandatory video decoder ~~codec~~ for the PSS. In addition, PSS should support:

- H.263 [**Error! Bookmark not defined.**] Profile 3 Level 10 decoder;
- MPEG-4 Visual Simple Profile Level 0 decoder, [**Error! Bookmark not defined.**] and [**Error! Bookmark not defined.**].

These two video decoders ~~codecs~~ are optional to implement.

NOTE: ITU-T Recommendation H.263 [**Error! Reference source not found.**] baseline has been mandated to ensure that video-enabled PSS support a minimum baseline video capability and interoperability can be guaranteed (an H.263 [**Error! Reference source not found.**] baseline bitstream can be decoded by both H.263 [**Error! Reference source not found.**] and MPEG-4 decoders). It also provides a simple upgrade path for mandating more advanced decoders ~~codecs~~ in the future (from both the ITU-T and ISO MPEG).

7.5 Still images

ISO/IEC JPEG [**Error! Bookmark not defined.**] together with JFIF [**Error! Bookmark not defined.**] decoders shall be supported. The support for ISO/IEC JPEG only apply to the following two modes:

- baseline DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF0' in [**Error! Bookmark not defined.**];
- progressive DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF2' [**Error! Bookmark not defined.**].

7.6 Bitmap graphics

The following bitmap graphics decoders ~~codecs~~ should be supported:

- GIF87a, [**Error! Bookmark not defined.**];
- GIF89a, [**Error! Bookmark not defined.**].

7.7 Vector graphics

No vector graphics decoder ~~codec~~ is specified for the simple PSS. For the extended PSS mandatory and/or optional vector graphics decoders ~~codecs~~ can be specified.

7.8 Text

The text decoder ~~codec~~ is intended to enable formatted text in a SMIL presentation. A PSS client shall support

- text formatted according to XHTML Basic [**Error! Bookmark not defined.**];

- rendering a SMIL presentation where text is referenced with the SMIL 2.0 "text" element together with the SMIL 2.0 "src" attribute.

The following character ~~en~~coding formats shall be supported:

- UTF-8, [**Error! Bookmark not defined.**];
- UCS-2, [**Error! Bookmark not defined.**].

NOTE: Since both SMIL and XHTML are XML based languages it would be possible to define a SMIL plus XHTML profile. In contrast to the present defined PSS4 SMIL Language Profile that only contain SMIL modules, such a profile would also contain XHTML modules. No combined SMIL and XHTML profile is specified for PSS. Rendering of such documents is out of the scope of the present document.