Technical Specification Group Services and System Aspects *TSGS#23 (04)0065*
Meeting #23, Phoenix, Arizona, USA, 15-18 March 2004

Source: TSG-SA WG4

Title: 3GPP TS 26.244 version 2.0.0 “3GPP file format (3GP)" (Release 6)

Agenda Item: 7.4.3

Presentation of Specification to TSG SA Plenary

Presentation to:	**TSG SA Meeting #23**
Document for presentation:	**TR 26.244 “3GPP file format (3GP)", Version 2.0.0**
Presented for:	**Approval**

Abstract of document:

The present document defines the 3GPP file format (3GP) of the 3GPP Transparent End-to-End Packet-switched Streaming Service (PSS) as an instance of the ISO base media file format. The definition addresses 3GPP specific features such as codec registration and conformance within the MMS and PSS services. A file format contains data in a structured way. The 3GPP file format can contain timing, structure and media data for multimedia streams. It is used by MMS and PSS for timed visual and aural multimedia.

Changes since last presentation:

This is version 2.0.0, provided for approval to TSG-SA#22 Plenary.

Outstanding Issues:

None.

Contentious Issues:

None.

3GPP TS 26.244 V2.0.0 (2004-03)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Services and System Aspects Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP) (Release 6)

Keywords

3GPP, 3GP, PSS

3GPP

Postal address

3GPP support office address

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

Internet

http://www.3gpp.org

Copyright Notification

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

z the third digit is incremented when editorial only changes have been incorporated in the document.

The 3GPP transparent end-to-end packet-switched streaming service (PSS) specification consists of six 3GPP TSs: 3GPP TS 22.233 [1], 3GPP TS 26.233 [2], 3GPP TS 26.234 [3], 3GPP TS 26.245 [4], 3GPP TS 26.246 [5] and the present document.

The TS 22.233 contains the service requirements for the PSS. The TS 26.233 provides an overview of the PSS. The TS 26.234 provides the details of protocol and codecs used by the PSS. The TS 26.245 defines the Timed text format used by the PSS. The TS 26.246 defines the 3GPP SMIL language profile. The present document defines the 3GPP file format (3GP) used by the PPS and MMS services.

The TS 26.244 (present document), TS 26.245 and TS 26.246 start with Release 6. Earlier releases of the 3GPP file format, the Timed text format and the 3GPP SMIL language profile can be found in TS 26.234.

Introduction

A file format contains data in a structured way. The 3GPP file format can contain timing, structure and media data for multimedia streams. It is used by MMS and PSS for timed visual and aural multimedia.

1 Scope

The present document defines the 3GPP file format (3GP) as an instance of the ISO base media file format. The definition addresses 3GPP specific features such as codec registration and conformance within the MMS and PSS services.

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 22.233: "Transparent End-to-End Packet-switched Streaming Service; Stage 1".

[2] 3GPP TS 26.233: "Transparent end-to-end packet switched streaming service (PSS); General description".

[3] 3GPP TS 26.234: "Transparent end-to-end packet switched streaming service (PSS); Protocols and codecs".

[4] 3GPP TS 26.245: "Transparent end-to-end packet switched streaming service (PSS); Timed text format".

[5] 3GPP TS 26.246: "Transparent end-to-end packet switched streaming service (PSS); 3GPP SMIL Language Profile".

[6] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[7] ISO/IEC 14496-12:2003 | 15444-12:2003: "Information technology – Coding of audio-visual objects – Part 12: ISO base media file format" | "Information technology – JPEG 2000 image coding system – Part 12: ISO base media file format".

[8] 3GPP TS 26.140: "Multimedia Messaging Service (MMS); Media formats and codecs".

[9] ITU-T Recommendation H.263 (1998): "Video coding for low bit rate communication".

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

[11] 3GPP TS 26.071: "Mandatory Speech CODEC speech processing functions; AMR Speech CODEC; General description".

[12] 3GPP TS 26.171: "AMR Wideband Speech Codec; General Description".

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

[14] ISO/IEC 14496-14:2003: "Information technology – Coding of audio-visual objects – Part 14: MP4 file format".

[15] IETF RFC 3267: "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", Sjoberg J. et al., June 2002.

[16] 3GPP TS 26.101: "Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec frame structure".

[17] 3GPP TS 26.201: "Speech Codec speech processing functions; AMR Wideband Speech Codec; Frame Structure".

[18] ITU-T Recommendation H.263 – Annex X (2001): "Annex X: Profiles and levels definition".

[19] IETF RFC 3711: "The Secure Real-time Transport Protocol", Baugher M. et al., Feb 2004.

[20] ISO/IEC 14496-15: "Information technology – Coding of audio-visual objects – Part 15: Advanced Video Coding (AVC) file format".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

continuous media: media with an inherent notion of time. In the present document speech, audio, video and timed text

discrete media: media that itself does not contain an element of time. In the present document all media not defined as continuous media

PSS client: client for the 3GPP packet switched streaming service based on the IETF RTSP/SDP and/or HTTP standards, with possible additional 3GPP requirements according to [3]

PSS server: server for the 3GPP packet switched streaming service based on the IETF RTSP/SDP and/or HTTP standards, with possible additional 3GPP requirements according to [3]

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [6] and the following apply.

3GP	3GPP file format
AAC	Advanced Audio Coding
BIFS	Binary Format for Scenes
ITU-T	International Telecommunications Union – Telecommunications
MIME	Multipurpose Internet Mail Extensions
MMS	Multimedia Messaging Service
MP4	MPEG-4 file format
PSS	Packet-switched Streaming Service
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SDP	Session Description Protocol
SRTP	Secure Real-time Transport Protocol

4 Overview

The 3GPP file format (3GP) is defined in this specification as an instance of the ISO base media file format [7]. It is mandated in [8] to be used for continuous media along the entire delivery chain envisaged by the MMS, independent on whether the final delivery is done by streaming or download, thus enhancing interoperability.

In particular, the following stages are considered:

- upload from the originating terminal to the MMS proxy;

- file exchange between MMS servers;

- transfer of the media content to the receiving terminal, either by file download or by streaming. In the first case the self-contained file is transferred, whereas in the second case the content is extracted from the file and streamed according to open payload formats. In this case, no trace of the file format remains in the content that goes on the wire/in the air.

For the PSS, the 3GPP file format is mandated in [3] to be used for timed text and it should be supported by PSS servers; 3GP files with streaming-server extensions should be used for storage in streaming servers and the "hint track" mechanism should be used for the preparation for streaming.

5 Conformance

5.1 General

The 3GPP file format is structurally based on the ISO base media file format defined in [7]. However, the conformance statement for 3GP files is defined here by addressing constraints and extensions to the ISO base media file format, registration of codecs, file identification (file extension, brand identifier and MIME type) and profiles. If a 3GP file contains codecs or functionalities not conforming to this specification they may be ignored, i.e. a 3GP compliant file parser may ignore non-compliant boxes.

5.2 Definition

5.2.1 Limitations to the ISO base media file format

The following limitations to the ISO base media file format [7] shall apply to a 3GP file:

- compact sample sizes ('stz2') shall not be used;
- movie fragments shall not be used.

5.2.2 Registration of codecs

Code streams for H.263 video [9], MPEG-4 video [10], AMR narrow-band speech [11], AMR wide-band speech [12], MPEG-4 AAC audio [13], and timed text [4] can be included in 3GP files as described in clause 6 of the present document.

5.2.3 Extensions

The following extensions to the ISO base media file format [7] can be used in a 3GP file:

- streaming-server extensions (see clause 7);
- asset information (see clause 8);
- video-buffer information (see clause 9);
- encryption (see clause 10).

If SDP information is included in a 3GP file, it shall be used as defined by the streaming-server extensions.

5.2.4 MPEG-4 systems specific elements

For the storage of MPEG-4 media specific information in 3GP files, this specification refers to MP4 [14], which is also based on the ISO base media file format. However, tracks relative to MPEG-4 system architectural elements (e.g. BIFS scene description tracks or OD Object descriptors) are optional in 3GP files and shall be ignored. The inclusion of MPEG-4 media does not imply the usage of MPEG-4 systems architecture. Terminals and servers are not required to implement any of the specific MPEG-4 system architectural elements.

5.2.5 Template fields

The ISO base media file format [7] defines the concept of template fields that may be used by derived file formats. The template field “alternate group” can be used in 3GP files, as defined in clause 7.2. No other template fields are used.

5.2.6 Interpretation of the 3GPP file format

All index numbers used in the 3GPP file format start with the value one rather than zero, in particular “first-chunk” in Sample to chunk box, “sample-number” in Sync sample box and “shadowed-sample-number”, “sync-sample-number” in Shadow sync sample box.

5.3 Identification

5.3.1 General

3GP files can be identified using several mechanisms: file extension, MIME types and brands.

5.3.2 File extension

When stored in traditional computer file systems, 3GP files should be given the file extension '.3gp'. Readers should allow mixed case for the alphabetic characters.

5.3.3 MIME types

The MIME types 'video/3gpp' (for visual or audio/visual content, where visual includes both video and timed text) and 'audio/3gpp' (for purely audio content) are expected to be registered and used.

5.3.4 Brands

This specification defines several brand identifiers corresponding to the profiles defined in clause 5.4. Brands are indicated in a file-type box, defined in [7], which shall be present in conforming files. The fields of the file-type box shall be used as follows:

- Brand: Identifies the ‘best use’ of the file and should match the file extension. For files with extension '.3gp' and conforming to this specification, the brand shall be one of the profile brands defined in clause 5.4.
- MinorVersion: This identifies the minor version of the brand. For files with brand '3gLZ', where L is a letter and Z a digit, and conforming to version Z.x.y of this specification, this field takes the value x*256 + y.
- CompatibleBrands: a list of brand identifiers (to the end of the box). Any profile of a 3GP file is declared by including the corresponding brand from clause 5.4 in this list.

The brand identifier (of one of the profiles) must occur in the compatible-brands list, and may also be the primary brand. Conformance to more than one profile is indicated by listing the corresponding brands in the compatible-brands list. If the file is also conformant to earlier releases of this specification, it is recommended that the corresponding brands ('3gp4' and/or '3gp5') also occur in the compatible-brands list. If '3gp4' or '3gp5' is not in the compatible-brands list, the file will not be processed by a Release 4 or Release 5 reader, respectively. Readers should check the compatible-brands list for the identifiers they recognize, and not rely on the file having a particular primary brand, 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.

5.4 Profiles

5.4.1 General

All 3GP files of Release 6 shall conform to the general definitions in clauses 5.1-5.3. Additional profile-specific constraints are listed below. A 3GP file must conform to at least one profile and may conform to several profiles.

5.4.2 General profile

The 3GP General profile is branded '3gg6' and is a superset of all other profiles. It is used to identify 3GP files conformant to this specification, although they may not conform to any of the specific profiles listed below.

NOTE: The General profile of 3GP have less restrictions than other profiles and is suitable for files not yet ready to be delivered by MMS or to be streamed by a PSS server. A General 3GP file may for instance contain several alternative tracks of media. After extracting a suitable set of tracks the file may be ready for MMS and can be re-profiled as a Basic file. Alternatively, by adding streaming-server extensions, it may be re-profiled as a Streaming-server profile.

5.4.3 Basic profile

The 3GP Basic profile is branded '3gp6' and is used in MMS and PSS. Conformance to this profile will guarantee the 3GPP file format to be used internally within the MMS service, as well as PSS to interwork with MMS.

The following constraints shall apply to a 3GP file conforming to Basic profile:

- there shall be no references to external media outside the file, i.e. a file shall be self-contained;
- the maximum number of tracks shall be one for video, one for audio and one for text;
- the maximum number of sample entries shall be one per track for video and audio (but unrestricted for text).

NOTE: The Basic profile of 3GP in Release 6 corresponds to 3GP files of earlier releases, which did not define profiles. Files with brands '3gp4' and '3gp5' in Release 4 and 5, respectively, correspond to files with brand '3gp6' in Release 6.

5.4.4 Streaming-server profile

The 3GP Streaming-server profile is branded '3gs6' and is used in PSS. Conformance to this profile will guarantee interoperability between content creation tools and streaming servers, in particular for the selection of alternative encodings of content and adaptation during streaming.

The following constraints shall apply to 3GP files conforming to Streaming-server profile:

- RTP hint tracks shall be included for all media tracks;
- RTP hint tracks shall comply with streaming as specified by PSS [3];
- SDP information shall be included, as specified in clause 7.5, where SDP fragments shall be stored in the hint tracks with media-level control URLs referring to (the same) hint tracks.
- streaming-server extensions should be used for hint tracks, as defined in chapter 7.

The following requirements shall apply to servers conforming to this profile. A conforming server

- shall understand and respect directions given in the streaming-server extensions, as defined in chapter 7;
- should understand hint tracks;
- may override instructions in hint tracks.

NOTE 1: The instructions given in RTP hint tracks shall be consistent with the PSS. In particular, send times of RTP packets shall respect buffer constraints and be consistent with parameters used in SDP.

NOTE 2: Earlier releases of the 3GPP file format did not define streaming-server extensions or profiles. The usage of hint tracks was an internal implementation matter for servers outside the scope of the PSS specification.

5.4.5 Progressive-download profile

The 3GP Progressive-download profile is branded '3gr6'. It is used to label 3GP files that are suitable for progressive download, i.e. a scenario where a file may be played during download (with some delay).

The following constraints shall apply to 3GP files conforming to Progressive-download profile:

- the ‘moov’ box shall be placed right after the ‘ftyp’ box in the beginning of the file;
- all media tracks (if more than one) shall be interleaved with an interleaving depth of one second or less.

NOTE 1: This profile functions as an aid and not a requirement for progressive download, which has been an inherent feature of the 3GPP file format since the first version in Release 4. By parsing a 3GP file, a client can always determine whether a file can be progressively downloaded, and then calculate the interleaving depth from the meta-data in the ‘moov’ box.

NOTE 2: The ‘interleaving depth of one second or less’ means that:

- Each chunk contains one or more samples, with the total duration of the samples being either: no greater than 1 second, or the duration of a single sample if that sample’s duration is greater than 1 second;
- Within a track, chunks must be in decoding time order within the media-data box ‘mdat’;
- It is recommended that, in ‘mdat’, regardless of media type, the chunks for all tracks are stored in ascending order by decoding time. However, this order may be perturbed so that, when two chunks from different tracks overlap in time, the chunk of one track (e.g. audio) is stored before the chunk of the other track (e.g. video), even if the first sample in the second track has a slightly earlier timestamp than the first sample in the first track.

5.5 File-branding guidelines

The file-type brands defined in this specification are used to label 3GP files belonging to Release 6 and conforming to one or more profiles. 3GP files may also conform to earlier Releases or even to other file formats, such as MP4, which is also derived from the ISO base media file format [7].

Table 5.1 contains a non-exhaustive list of examples with 3GP files for various purposes. All 3GP files of Release 5 or later shall contain the compatible brand ‘isom’ indicating that they conform to the ISO base media file format. The major brand shall be included in the compatible brands list as well. If a file contains more than one (3GPP) brand in the compatible brands list, the major brand indicates the “best use” of the file. For example, a Release-5 file with audio combined with Timed text is best played by a Release-5 player, but may also be played by a Release-4 player that does not support timed text.

Table 5.1: Examples of brand usage in 3GP files

Conformance	Suffix	Brand	Compatible brands	Example content
MMS and download: Files shall contain one or more of the brands 3gp4, 3gp5 and 3gp6. It is good practice to include compatible brands of earlier releases to enable legacy players to play the files.				
Release 4	.3gp	3gp4	3gp4	H.263 and AMR
Release 5, 4	.3gp	3gp5	3gp5, 3gp4, isom	H.263 and AMR
Release 6, 5, 4	.3gp	3gp6	3gp6, 3gp5, 3gp4, isom	H.263 and AMR
Release 6, 5, 4	.3gp	3gp6	3gp6, 3gp5, 3gp4, isom	H.263, AMR and Timed text
Release 6, 5	.3gp	3gp6	3gp6, 3gp5, isom	Timed text
Release 6	.3gp	3gp6	3gp6, isom	Some Release-6 specific codec TBD
Progressive download and MMS				
Release 6, 5, 4	.3gp	3gr6	3gr6, 3gp6, 3gp5, 3gp4, isom	H.263
Release 6, 5, 4	.3gp	3gr6	3gr6, 3gp6, 3gp5, 3gp4, isom	interleaved H.263 and AMR
Streaming servers: Some files may in principle also be used for MMS or download.				
Release 6	.3gp	3gs6	3gs6, isom	AMR and hint track
Release 6	.3gp	3gs6	3gs6, isom	2 tracks H.263 and 2 hint tracks
Release 6, 5, 4	.3gp	3gs6	3gs6, 3gp6, 3gp5, 3gp4, isom	H.263, AMR and hint tracks
General purpose: Files that are not yet suitable for MMS, download or PSS streaming servers.				
Release 6	.3gp	3gg6	3gg6, isom	4 tracks H.263 (and no hint tracks)
Release 6	.3gp	3gg6	3gg6, isom	2 tracks H.263, 3 tracks AMR
3GP file, also conforming to MP4				
Release 4, 5 and MP4	.3gp	3gp5	3gp5, 3gp4, mp42, isom	MPEG-4 video
MP4 file, also conforming to 3GP				
Release 5 and MP4	.mp4	mp42	mp42, 3gp5, isom	MPEG-4 video and AAC

6 Codec registration

6.1 General

The purpose of this clause is to define the necessary structure for integration of the H.263, MPEG-4 video, AMR, AMR-WB, and AAC media specific information in a 3GP file. Clause 6.2 gives some background information about the Sample Description box in the ISO base media file format [7] and clauses 6.3 and 6.4 about the MP4VisualSampleEntry box and the MP4AudioSampleEntry box in the MPEG-4 file format [14]. The definitions of the Sample Entry boxes for AMR, AMR-WB and H.263 are given in clauses 6.5 to 6.8. The integration of timed text in a 3GP file is specified in [4].

AMR and AMR-WB data is stored in the stream according to the AMR and AMR-WB storage format for single channel header of Annex E [15], without the AMR magic numbers.

6.2 Sample Description box

In an ISO file, Sample Description Box gives detailed information about the coding type used, and any initialisation information needed for that coding. The Sample Description Box can be found in the ISO file format Box Structure Hierarchy shown in figure 6.1.

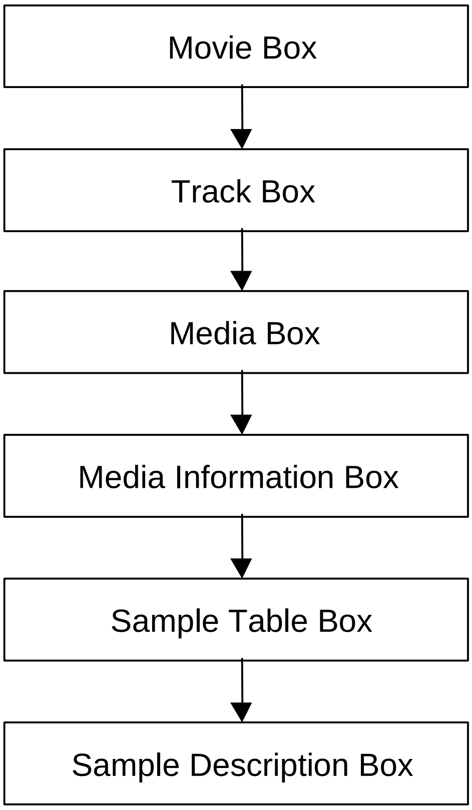

Figure 6.1: ISO File Format Box Structure Hierarchy

The Sample Description Box can have one or more Sample Entries. Valid Sample Entries already defined for ISO and MP4 include MP4AudioSampleEntry, MP4VisualSampleEntry and HintSampleEntry. The Sample Entries for AMR and AMR-WB shall be AMRSampleEntry, for H.263 it shall be H263SampleEntry, and for timed text it shall be TextSampleEntry.

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

```
SampleEntry ::=    MP4VisualSampleEntry |
                   MP4AudioSampleEntry |
                   AMRSampleEntry |
```

H263SampleEntry |
TextSampleEntry |
HintSampleEntry

Table 6.1: SampleEntry fields

Field	Type	Details	Value
MP4VisualSampleEntry		Entry type for visual samples defined in the MP4 specification.	
MP4AudioSampleEntry		Entry type for audio samples defined in the MP4 specification.	
AMRSampleEntry		Entry type for AMR and AMR-WB speech samples defined in clause 6.5 of the present document.	
H263SampleEntry		Entry type for H.263 visual samples defined in clause 6.6 of the present document.	
TextSampleEntry		Entry type for timed text samples defined in the timed text specification	
HintSampleEntry		Entry type for hint track samples defined in the ISO specification.	

From the above 6 Sample Entries, only the MP4VisualSampleEntry, MP4AudioSampleEntry, H263SampleEntry and AMRSampleEntry are taken into consideration here. TextSampleEntry is defined in [4] and HintSampleEntry in [7].

6.3 MP4VisualSampleEntry box

The MP4VisualSampleEntry Box is defined as follows:

MP4VisualSampleEntry ::= BoxHeader
Reserved_6
Data-reference-index
Reserved_16
Width
Height
Reserved_4
Reserved_4
Reserved_4
Reserved_2
Reserved_32
Reserved_2
Reserved_2
ESDBox

Table 6.2: MP4VisualSampleEntry fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.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 boxes.	
Reserved_16	Const unsigned int(32) [4]		0
Width	Unsigned int(16)	Maximum width, in pixels of the stream	
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
ESDBox		Box containing an elementary stream descriptor for this stream.	

The stream type specific information is in the ESDBox structure, as defined in [14].

This version of the MP4VisualSampleEntry, 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.

6.4 MP4AudioSampleEntry box

MP4AudioSampleEntryBox is defined as follows:

MP4AudioSampleEntry ::= BoxHeader
Reserved_6
Data-reference-index
Reserved_8
Reserved_2
Reserved_2
Reserved_4
TimeScale
Reserved_2
ESDBox

Table 6.3: MP4AudioSampleEntry fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.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 boxes.	
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
ESDBox		Box containing an elementary stream descriptor for this stream.	

The stream type specific information is in the ESDBox structure, as defined in [14].

6.5 AMRSampleEntry box

For narrow-band AMR, the box type of the AMRSampleEntry Box shall be 'samr'. For AMR wideband (AMR-WB), the box type of the AMRSampleEntry Box shall be 'sawb'.

The AMRSampleEntry Box is defined as follows:

AMRSampleEntry ::= **BoxHeader**
Reserved_6
Data-reference-index
Reserved_8
Reserved_2
Reserved_2
Reserved_4
TimeScale
Reserved_2
AMRSpecificBox

Table 6.4: AMRSampleEntry fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.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 boxes.	
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 box of this media	
Reserved_2	Const unsigned int(16)		0
AMRSpecificBox		Information specific to the decoder.	

If one compares the MP4AudioSampleEntry Box - AMRSampleEntry Box the main difference is in the replacement of the ESDBox, which is specific to MPEG-4 systems, with a box suitable for AMR and AMR-WB. The **AMRSpecificBox** field structure is described in clause 6.7.

6.6 H263SampleEntry box

The box type of the H263SampleEntry Box shall be 's263'.

The H263SampleEntry Box is defined as follows:

H263SampleEntry ::= **BoxHeader**
Reserved_6
Data-reference-index
Reserved_16
Width
Height
Reserved_4
Reserved_4
Reserved_4
Reserved_2
Reserved_32
Reserved_2
Reserved_2
H263SpecificBox

Table 6.5: H263SampleEntry fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.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 boxes.	
Reserved_16	Const unsigned int(32) [4]		0
Width	Unsigned int(16)	Maximum width, in pixels of the stream	
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
H263SpecificBox		Information specific to the H.263 decoder.	

If one compares the MP4VisualSampleEntry – H263SampleEntry Box the main difference is in the replacement of the ESDBox, which is specific to MPEG-4 systems, with a box suitable for H.263. The **H263SpecificBox** field structure for H.263 is described in clause 6.8.

6.7 AMRSpecificBox field for AMRSampleEntry box

The AMRSpecificBox fields for AMR and AMR-WB shall be as defined in table 6.6. The AMRSpecificBox for the AMRSampleEntry Box shall always be included if the 3GP file contains AMR or AMR-WB media.

Table 6.6: The AMRSpecificBox fields for AMRSampleEntry

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		‘damr’
DecSpecificInfo	AMRDecSpecStruc	Structure which holds the AMR and AMR-WB Specific information	

BoxHeader Size and Type: indicate the size and type of the AMR decoder-specific box. 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-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.

mode_set: the active codec modes. 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 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 [16]. 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 [17]. 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 Modes 0, 2, 4, 7 and 8 are present in the 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 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 3GP file. This number shall be greater than 0 and less than 16. A value of 1 means each frame is treated as one sample. A value of 10 means that 10 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 stream, the number of frames can be smaller than frames_per_sample, if the number of remaining frames is smaller than frames_per_sample.

NOTE1: The "hinter", for the creation of the hint tracks, can use the information given by the AMRDecSpecStruc members.

NOTE2: The following AMR MIME parameters are not relevant to PSS: {mode_set, mode_change_period, mode_change_neighbor}. PSS servers should not send these parameters in SDP, and PSS clients shall ignore these parameters if received.

6.8 H263SpecificBox field for H263SampleEntry box

The H263SpecificBox fields for H. 263 shall be as defined in table 6.7. The H263SpecificBox for the H263SampleEntry Box shall always be included if the 3GP file contains H.263 media.

The H263SpecificBox for H263 is composed of the following fields.

Table 6.7: The H263SpecificBox fields H263SampleEntry

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		‘d263’
DecSpecificInfo	H263DecSpecStruc	Structure which holds the H.263 Specific information	
BitrateBox		Specific bitrate information (optional)	

BoxHeader Size and Type: indicate the size and type of the H.263 decoder-specific box. 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
}
```

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.

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 [18].

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}

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

The BitrateBox field shall be as defined in table 6.8. The BitrateBox may be included if the 3GP file contains H.263 media.

The BitrateBox is composed of the following fields.

Table 6.8: The BitrateBox fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		‘bitr’
DecBitrateInfo	DecBitrStruc	Structure which holds the Bitrate information	

BoxHeader Size and Type: indicate the size and type of the bitrate box. The type must be ‘bitr’.

DecBitrateInfo: This is the structure where the stream bitrate information resides.

DecBitrStruc is defined as follows:

```
struct DecBitrStruc{
                    Unsigned int (32)    Avg_Bitrate
                    Unsigned int (32)    Max_Bitrate
}
```

The definitions of DecBitrStruc members are as follows:

Avg_Bitrate: the average bitrate in bits per second of this elementary stream. For streams with variable bitrate this value shall be set to zero.

Max_Bitrate: the maximum bitrate in bits per second of this elementary stream in any time window of one second duration.

7 Streaming-server extensions

7.1 General

This clause defines extensions to 3GP files to be used by streaming servers. The extensions enable a PSS server to relate different tracks and use them for selection and adaptation. In particular, they enable a PSS server to

- generate SDP descriptions with alternatives, as specified in subclauses 5.3.3.3 - 5.3.3.4 of [3];
- select and combine tracks with alternative encodings of media before a presentation;
- switch between tracks with alternative encodings during a streaming session.

The streaming-server extensions are intended to be used with hint tracks, although they are not limited to be used with hint tracks. Hint tracks are defined in the ISO base media file format [7] and provide (RTP) packetization instructions for media stored in a file.

NOTE: The present document defines syntax and semantics for streaming-server extensions in 3GP files. It does not define protocols for, e.g., how a PSS server signals alternative encodings or switches between different bitrate encodings. All protocols used by a PSS server are defined in [3].

7.2 Groupings of alternative tracks

By default all enabled tracks in a 3GP file are streamed (played) simultaneously. However, the ISO base media file format [7] specifies that tracks that are alternatives to each other can be grouped into an alternate group. Tracks in an alternate group that can be used for switching can be further grouped into a switch group, as defined here.

7.2.1 Alternate group

Alternate group is encoded as an integer in the Track Header box of each track. If this integer is 0 (default value), there is no information on possible relations to other tracks. If this integer is not 0, it should be the same for tracks that contain alternate data for one another and different for tracks belonging to different such groups. Only one track within an alternate group should be streamed or played at any time and must be distinguishable from other tracks in the group via attributes such as bitrate, codec, language, packet size etc.

7.2.2 Switch group

Switch group is encoded as an integer in the Track Selection box of each track, as defined below. If this box is absent or if this integer is 0 (default value), there is no information on whether the track can be used for switching during streaming or playing. If this integer is not 0, it shall be the same for tracks that can be used for switching between each other. Tracks that belong to the same switch group shall belong to the same alternate group.

7.3 Track Selection box

This subclause defines an optional box that aids the selection between tracks. It is used to encode switch groups and the criteria that should be used to differentiate tracks within alternate and switch groups.

The Track Selection box is defined in table 7.1. It is contained in the User data box of the track it modifies.

Table 7.1: Track Selection box fields

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'tsel'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
SwitchGroup	int(32)	Switch group of track.	0 (default)
AttributeList	Unsigned int(32) [N]	List of N attributes to the end of the box.	

BoxHeader Size, Type, Version and Flags: indicate the size, type, version and flags of the Track Selection box. The type shall be 'tsel' and the version shall be 0. No flags are defined.

SwitchGroup: indicates switch group as defined in clause 7.2.2. It shall be 0 if the track is not intended for switching.

AttributeList: is a list of attributes to the end of the box. The attributes in this list should be used as differentiation criteria for tracks in the same alternate or switch group. Each attribute is associated with a pointer to the field or information that distinguishes the track. Attributes and pointers are listed in table 7.2.

Table 7.2: Attributes for AttributeList of the Track Selection box

Name	Attribute	Pointer
Language	'lang'	Value of grouping type LANG of "alt-group" attribute in session-level SDP (defined in clause 5.3.3.4 of [3])
Bandwidth	'bwas'	Value of "b=AS" attribute in media-level SDP
Codec	'cdec'	SampleEntry (in Sample Description box of media track)
Screen size	'scsz'	Width and height fields of MP4VisualSampleEntry and H263SampleEntry (in media track)
Max packet size	'mpsz'	Maxpacketsize field in RTPHintSampleEntry
Media type	'mtyp'	Handlertype in Handler box (of media track)

7.4 Combining alternative tracks

Tracks from different alternate groups are streamed (played) simultaneously. However, all combinations of tracks may not form suitable presentations. In order to suggest suitable combinations of tracks and also to reduce the number of possible combinations, a content provider can encode preferred combinations of alternative tracks in a 3GP file. Such combinations are encoded by the "alt-group" attribute in the session-level SDP fragment, as described in clause 7.5.3.

If information on suitable combinations of tracks is missing, tracks with the lowest track IDs of each alternate group should be streamed (played) by default.

7.5 SDP

7.5.1 Session- and media-level SDP

Fragments that together constitute an SDP description shall be contained in a 3GP file with streaming-server extensions. Session-level SDP, i.e. all lines before the first media-specific line ("m=" line), shall be stored as Movie SDP information within the User Data box, as specified in [7]. Media-level SDP, i.e. an "m=" line and the lines before the next "m=" line (or end of SDP) shall be stored as Track SDP information within the User data box of the corresponding track. Media-level SDP shall be contained in hint tracks (if provided).

7.5.2 Stored versus generated SDP fields

The SDP information stored in a 3GP file should be as complete as possible, although some fields must be generated or modified by the server when a presentation is composed. Table 7.3 gives an overview of the SDP fields used by PSS, c.f. Table A.1 in [3], and whether they are required to be included in 3GP files or whether the server is required to generate them.

Table 7.3: Overview of stored and generated fields in SDP

Type	Description		Contained in 3GP file	Generated by PSS server
Session Description				
V	Protocol version		R	O
O	Owner/creator and session identifier		O	R
S	Session Name		R	O
I	Session information		O	O
U	URI of description		O	O
E	Email address		O	O
P	Phone number		O	O
C	Connection Information		O	R
B	Bandwidth information	AS	O	O
		RS	O	O
		RR	O	O
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	R	O
		alt-group	R (see note 4)	O
		QoE-Metrics	O	O
		3GPP-Asset-Information	O	O
One or more Media Descriptions (See below)				
Time Description				
T	Time the session is active		R	O
R	Repeat times		O	O
Media Description				
M	Media name and transport address		R	O
I	Media title		O	O
C	Connection information		O	R
B	Bandwidth information	AS	R	O
		RS	O	R
		RR	O	R
K	Encryption Key		O	O
A	Attribute Lines	control	O	R
		range	R	O
		fmtp	R	O
		rtpmap	R	O
		X-predecbufsize	R (see note 5)	O
		X-initpredecbufperiod	R (see note 5)	O
		X-initpostdecbufperiod	R (see note 5)	O
		X-decbyterate	R (see note 5)	O
		framesize	R	O
		alt	N	R
		alt-default-id	N	R
		3GPP-Adaptation-Support	N	O
		QoE-Metrics	O	O
		3GPP-Asset-Information	O	O

Note 1: Fields in 3GP files are Required (R), Optional (O), or Not allowed (N).

Note 2: Servers are Required (R) to generate (possibly by copying or modifying from file), or have the Option (O) to generate/copy/modify, or are Not allowed (N) to modify fields. If a field is present in a file, it shall be copied or modified, but not omitted, by the server.

Note 3: Some types shall only be included under certain conditions, as specified by PSS [3].

Note 4: The “alt-group” attribute is required to be stored in 3GP files if it is used.

Note 5: The "X-" attributes are required to be stored in 3GP files if they are used. They may either be specified in the PSS Annex G box '3gag' (see Clause 9) or in media-level SDP fragments.

7.5.3 SDP attributes for alternatives

Clauses 5.3.3.3 and 5.3.3.4 of [3] define SDP attributes that a server can use for presenting options to a client. These attributes can be used to encode suggested groupings of tracks, e.g. for selecting a certain language or target bitrate.

Suggested groupings of tracks from different alternate groups, i.e. groupings of tracks that should be streamed together, are encoded by using the “alt-group” attribute in the session-level SDP. Note that a server may have to prune options from such groupings if certain tracks are not presented to the client.

Media-level SDP fragments shall not contain alternative-media attributes (“alt” and “alt-default-id”) as they are difficult to pre-encode. When the server combines several media-level SDP fragments from alternative tracks into one media-level SDP, it must generate the appropriate “alt” and “alt-default-id” attributes. This can be done by using the information provided in the “alt-group” attributes in the session-level SDP.

NOTE 1: Track IDs given by the Track Header boxes shall be used for alternative IDs (“alt-id”) in attributes for SDP alternatives.

NOTE 2: Tracks with the lowest track IDs of each alternate group should be used as default tracks, i.e. used with the “alt-default-id” attributes.

7.6 SRTP

Hinted content may require the use of SRTP [19] for streaming, e.g. for integrity protection, by using the hint-track format for SRTP defined here. It consists of a dedicated sample entry, which will be ignored by 3GP servers not capable of handling SRTP.

SRTP hint tracks are formatted identically to RTP hint tracks defined in [7], except that:

- the sample entry name is changed from 'rtp ' to 'srtp' to indicate to the server that SRTP is required;
- an extra box is added to the sample entry which can be used to instruct the server in the nature of the on-the-fly encryption and integrity protection that must be applied.

Samples of an SRTP hint track follow the same syntax for constructing RTP packets as RTP hint tracks.

An SRTP Hint Sample Entry ('srtp') shall include an SRTP Process Box ('srpp') that may instruct the server as to which SRTP algorithms should be applied. It is defined in Table 7.4.

Table 7.4: SRTPProcessBox

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'srpp'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
EncryptionAlgorithmRTP	Unsigned int(32)	4cc identifying the algorithm	
EncryptionAlgorithmRTCP	Unsigned int(32)	4cc identifying the algorithm	
IntegrityAlgorithmRTP	Unsigned int(32)	4cc identifying the algorithm	
IntegrityAlgorithmRTCP	Unsigned int(32)	4cc identifying the algorithm	
SchemeTypeBox		Box containing the protection scheme.	
SchemeInformationBox		Box containing the scheme information.	

The **SchemeTypeBox** and **SchemeInformationBox** have the syntax defined in Tables 10.7 and 10.8, respectively. They serve to provide the parameters required for applying SRTP. The Scheme Type Box is used to indicate the necessary key management and security policy for the stream in extension to the defined algorithmic pointers provided by the SRTP Process Box. The key management functionality is also used to establish all the necessary SRTP parameters. The key management functionality is also used to establish all the necessary SRTP parameters as listed in section 8.2 of [19]. The exact definition of protection schemes is out of the scope of the file format.

The algorithms for encryption and integrity protection are defined by SRTP. Table 7.5 summarizes the format identifiers defined here. An entry of four spaces ($20$20$20$20) may be used to indicate that a process outside the file format decides the choice of algorithm for either encryption or integrity protection.

Table 7.5: Algorithms for encryption and integrity protection

Format	Algorithm
$20$20$20$20	The choice of algorithm for either encryption or integrity protection is decided by a process outside the file format
ACM1	Encryption using AES in Counter Mode with 128-bit key, as defined in Section 4.1.1 of [19]
AF81	Encryption using AES in F8-mode with 128-bit key, as defined in Section 4.1.2 of [19]
ENUL	Encryption using the NULL-algorithm as defined in Section 4.1.3 of [19]
SHM2	Integrity protection using HMAC-SHA-1 with 160-bit key, as defined in Section 4.2.1 of [19]
ANUL	Integrity protection not applied to RTP (but still applied to RTCP). Note: this is valid only for IntegrityAlgorithmRTP.

8 Asset information

A user-data box ('udta'), as defined in [7] may be present in conforming files. It should reside within the Movie box, but may reside within the Track box, following the hierarchy of boxes described in Clause 6.2.

Within the user-data box, there may reside sub-boxes that contain asset meta-data, taken from the list of boxes in tables 8.1 through 8.10 below (zero or more sub-boxes of each kind, zero or one for each language or role of location information). Each of the sub-boxes conforms to the definition of a "full box" as specified in [7] (hence the 'Version' and 'Flags' fields).

The following sub-boxes are in use for the following purposes:

- titl – title for the media (see table 8.1)
- dscp – caption or description for the media (see table 8.2)
- cprt – notice about organisation holding copyright for the media file (see table 8.3)
- perf – performer or artist (see table 8.4)
- auth – author of the media (see table 8.5)

- gnre – genre (category and style) of the media (see table 8.6)
- rtng – media rating (see table 8.7)
- clsf – classification of the media (see table 8.8)
- kywd – media keywords (see table 8.9)
- loci – location information (see table 8.10)

Table 8.1: The Title box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'titl'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Title	String	Text of title	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

Title: null-terminated string in either UTF-8 or UTF-16 characters, giving a title information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.2: The Description box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'dscp'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Description	String	Text of description	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

Description: null-terminated string in either UTF-8 or UTF-16 characters, giving a description information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.3: The Copyright box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'cprt'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Copyright	String	Text of copyright notice	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

Copyright: null-terminated string in either UTF-8 or UTF-16 characters, giving a copyright information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.4: The Performer box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'perf'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Performer	String	Text of performer	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

Performer: null-terminated string in either UTF-8 or UTF-16 characters, giving a performer information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.5: The Author box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'auth'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Author	String	Text of author	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

Author: null-terminated string in either UTF-8 or UTF-16 characters, giving an author information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.6: The Genre box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'gnre'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Genre	String	Text of genre	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

Genre: null-terminated string in either UTF-8 or UTF-16 characters, giving a genre information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.7: The Rating box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'rtng'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
RatingEntity	Unsigned int(32)	Four-character code rating entity	
RatingCriteria	Unsigned int(32)	Four-character code rating criteria	
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
RatingInfo	String	Text of media-rating information	

RatingEntity: four-character code that indicates the rating entity grading the asset, e.g., 'BBFC'. The values of this field should follow common names of worldwide movie rating systems, such as those mentioned in [http://www.movie-ratings.net/, October 2002].

RatingCriteria: four-character code that indicates which rating criteria are being used for the corresponding rating entity, e.g., ‘PG13’.

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

RatingInfo: null-terminated string in either UTF-8 or UTF-16 characters, giving a rating information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.8: The Classification box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'clsf'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
ClassificationEntity	Unsigned int(32)	Four-character code classification entity	
ClassificationTable	Unsigned int(16)	Index to classification table	
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
ClassificationInfo	String	Text of media-classification information	

ClassificationEntity: four-character code that indicates the classification entity classifying the asset. The values of this field should follow names of worldwide classification systems to be identified, but may be assigned blanks to indicate no specific classification entity.

ClassificationTable: binary code that indicates which classification table is being used for the corresponding classification entity. 0x00 is reserved to indicate no specific classification table.

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

ClassificationInfo: null-terminated string in either UTF-8 or UTF-16 characters, giving a classification information, taken from the corresponding classification table, if specified. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.9: The Keywords box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'kywd'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
KeywordCnt	Unsigned int(8)	Binary number of keywords	
Keywords	KeywordStruct[KeywordCnt]	Array of structures that hold the actual keywords (see Table 8.9.1)	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

KeywordCnt: binary code that indicates the number of keywords provided. This number shall be greater than 0.

Keywords: Array of structures that hold the actual keywords, according to table 8.9.1.

Table 8.9.1: The Keyword Struct

Field	Type	Details	Value
KeywordSize	Unsigned int(8)	Binary size of keyword	
KeywordInfo	String	Text of keyword	

KeywordSize: binary code that indicates the total size (in bytes) of the keyword information field.

KeywordInfo: null-terminated string in either UTF-8 or UTF-16 characters, giving a keyword information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Table 8.10: The Location Information box

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'loci'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
Pad	Bit(1)		0
Language	Unsigned int(5)[3]	Packed ISO-639-2/T language code	
Name	String	Text of place name	
Role	Unsigned int(8)	Non-negative value indicating role of location	
Longitude	Unsigned int(32)	Fixed-point value of the longitude	
Latitude	Unsigned int(32)	Fixed-point value of the latitude	
Altitude	Unsigned int(32)	Fixed-point value of the Altitude	
Astronomical_body	String	Text of astronomical body	
Additional_notes	String	Text of additional location-related information	

Language: declares the language code for the following text. See ISO 639-2/T for the set of three character codes. Each character is packed as the difference between its ASCII value and 0x60. The code is confined to being three lower-case letters, so these values are strictly positive.

Name: null-terminated string in either UTF-8 or UTF-16 characters, indicating the name of the place. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Role: indicates the role of the place. Value 0 indicates “shooting location”, 1 indicates “real location”, and 2 indicates “fictional location”. Other values are reserved.

Longitude: fixed-point 16.16 number indicating the longitude in degrees. Negative values represent western longitude.

Latitude: fixed-point 16.16 number indicating the latitude in degrees. Negative values represent southern latitude.

Altitude: fixed-point 16.16 number indicating the altitude in meters. The reference altitude, indicated by zero, is set to the sea level.

Astronomical_body: null-terminated string in either UTF-8 or UTF-16 characters, indicating the astronomical body on which the location exists, e.g. “earth”. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

Additional_notes: null-terminated string in either UTF-8 or UTF-16 characters, containing any additional location-related information. If UTF-16 is used, the string shall start with the BYTE ORDER MARK (0xFEFF).

NOTE 1: If the location information refers to a time-variant location, 'Name' should express a high-level location, such as “Finland” for several places in Finland or “Finland-Sweden” for several places in Finland and Sweden. Further details on time-variant locations can be provided as 'Additional notes'.

NOTE 2: The values of longitude, latitude and altitude provide cursory Global Positioning System (GPS) information of the media content.

NOTE 3: A value of longitude (latitude) that is less than –180 (-90) or greater than 180 (90) indicates that the GPS coordinates (longitude, latitude, altitude) are unspecified, i.e. none of the given values for longitude, latitude or altitude are valid.

9 Video buffer information

9.1 General

A 3GP file can include video-buffer parameters of the PSS buffering model, defined in Annex G of TS 26.234 [3] (PSS Annex G), that an associated video stream conforms to. For the case when only one set of parameters is associated to an entire video stream, these can be included in the corresponding media-level SDP fragment. However, in order to provide buffer parameters for different operation points, as defined below, and for different synchronization points, a track can contain a 3GPP PSS Annex G sample grouping as defined in this clause.

9.2 3GPP PSS Annex G sample grouping

A sample grouping is an assignment of each sample in a track to be a member of one sample group, based on a grouping criterion. The assignment of buffer parameters to synchronization points provides one sample grouping of all samples in a track. The usage of sample groups in 3GP files shall follow the syntax defined in [20].

The grouping type '3gag' defines the grouping criterion for 3GPP PSS Annex G buffer parameters. Zero or one sample-to-group box ('sbgp') for the grouping type '3gag' can be contained in the sample table box ('stbl') of a track. It shall reside in a hint track, if a hint track is used, otherwise in the video track. The presence of this box and grouping type indicates that the associated video stream complies with PSS Annex G. Note that the nature of the track defines the media transport for which the buffer parameters are calculated, e.g. for an RTP hint track, the media transport is RTP.

Each sample group is associated to zero or one sample group entries in the sample group description box ('sgpd'). Sample group entries for sample groups defined by the grouping type '3gag' are given by the 3GPP PSS Annex G Sample group entry defined in Table 9.1. Such a sample entry provides buffer parameters relevant to all samples in the corresponding sample group(s). Note that samples that are not synchronization points shall not be associated with a sample group.

Table 9.1: 3GPP PSS Annex G sample group entry

Field	Type	Details	Value
BufferParameters	AnnexGstruc	Structure which holds the buffer parameters of PSS Annex G	

BufferParameters: the structure where the PSS Annex G buffer parameters reside.

AnnexGstruc is defined as follows:

```
struct AnnexGstruc{
                    Unsigned int(16)    operation_point_count
                    for (i = 0; i < operation_point_count; i++){
                        Unsigned int (32)   tx_byte_rate
                        Unsigned int (32)   dec_byte_rate
                        Unsigned int (32)   pre_dec_buf_size
                        Unsigned int (32)   init_pre_dec_buf_period
                        Unsigned int (32)   init_post_dec_buf_period
                    }
}
```

The definitions of the AnnexGstruc members are as follows:

operation_point_count: specifies the number of operation points, each characterized by a pair of transmission byte rate and decoding byte rate. Values of buffering parameters are specified separately for each operation point. The value of operation_point_count shall be greater than 0.

tx_byte_rate: indicates the transmission byte rate (in bytes per second) that is used to calculate the transmission timestamps of media-transport packets for the PSS Annex G buffering verifier as follows. Let t1 be the transmission time of the previous media-transport packet and size1 be the number of bytes in the payload of the previous media-transport packet in transmission order, excluding the media-transport payload header and any lower-layer headers. For the first media-transport packet of the stream, t1 and size1 are equal to 0. The media track shall comply with PSS Annex G when each sample is packetized in one media-transport packet, the transmission order of media-transport packets is the same as their decoding order, and the transmission time of an media-transport packet is equal to t1 + size1 / tx_byte_rate. The value of tx_byte_rate shall be greater than 0.

dec_byte_rate: indicates the peak decoding byte rate that was used in this operation point to verify the compatibility of the stream with PSS Annex G. Values are given in bytes per second. The value of dec_byte_rate shall be greater than 0.

pre_dec_buf_size: indicates the size of the PSS Annex G hypothetical pre-decoder buffer in bytes that guarantees pauseless playback of the entire stream under the assumptions of PSS Annex G.

init_pre_dec_buf_period: indicates the required initial pre-decoder buffering period that guarantees pauseless playback of the entire stream under the assumptions of PSS Annex G. Values are interpreted as clock ticks of a 90-kHz block. That is, the value is incremented by one for each 1/90 000 seconds. For example, value 180 000 corresponds to a two second initial pre-decoder buffering.

init_post_dec_buf_period: indicates the required initial post-decoder buffering period that guarantees pauseless playback of the entire stream under the assumptions of PSS Annex G. Values are interpreted as clock ticks of a 90-kHz clock.

10 Encryption

10.1 General

A 3GP file may include encrypted media together with information on key management and requirements for decrypting and/or serving encrypted media. Tracks containing encrypted media use dedicated sample entries for encrypted media, which will be ignored by 3GP readers not capable of handling encrypted media. 3GP readers capable of detecting encrypted media are able to obtain "in the clear" the sample entries that apply to the decrypted media as well as all requirements for decrypting the media.

10.2 Sample entries for encrypted media tracks

The sample entries stored in the sample description box of a media track in a 3GP file identify the format of the encoded media, i.e. codec and other coding parameters. All valid sample entries for unencrypted media in a 3GP file are described in Clause 6. The principle behind storing encrypted media in a track is to "disguise" the original sample entry with a generic sample entry for encrypted media. Table 10.1 gives an overview of the formats (identifying sample entries) that can be used in 3GP files for signalling encrypted video, audio and text.

Table 10.1: Formats for encrypted media tracks

Format	Original format	Media content
'encv'	's263', 'mp4v', …	encrypted video: H.263, MPEG-4 visual, …
'enca'	'samr', 'sawb', 'mp4a', …	encrypted audio: AMR, AMR-WB, AAC, …
'enct'	'tx3g', …	encrypted text: timed text, …

The generic sample entries for encrypted media replicate the original sample entries and include a Protection scheme information box with details on the original format, as well as all requirements for decrypting the encoded media. The EncryptedVideoSampleEntry and the EncryptedAudioSampleEntry are defined in Tables 10.2 and 10.3, where the ProtectionSchemeInfoBox (defined in clause 10.2) is simply added to the list of boxes contained in a sample entry.

Table 10.2: EncryptedVideoSampleEntry

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'encv'
All fields and boxes of a visual sample entry, e.g. MP4VisualSampleEntry or H263SampleEntry.			
ProtectionSchemeInfoBox		Box with information on the original format and encryption	

Table 10.3: EncryptedAudioSampleEntry

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'enca'
All fields and boxes in an audio sample entry, e.g. MP4AudioSampleEntry or AMRSampleEntry.			
ProtectionSchemeInfoBox		Box with information on the original format and encryption	

The EncryptedVideoSampleEntry and the EncryptedAudioSampleEntry can also be used with any additional codecs added to the 3GP file format, as long as their sample entries are based on the SampleEntry of the ISO base media file format [7].

The EncryptedTextSampleEntry is defined in Table 10.4. Text tracks are specific to 3GP files and defined by the Timed text format [4]. In analogy with the cases for audio and video, a ProtectionSchemeInfoBox is added to the list of contained boxes.

Table 10.4: EncryptedTextSampleEntry

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'enct'
All fields and boxes of TextSampleEntry.			
ProtectionSchemeInfoBox		Box with information on the original format and encryption	

NOTE: The boxes within the sample entries defined in Tables 10.2-10.4 may not precede any of the fields. The order of the boxes (including the ProtectionSchemeInfoBox) is not important though.

10.3 Key management

The necessary requirements for decrypting media are stored in the Protection scheme information box. It contains the Original format box, which identifies the codec of the decrypted media, the Scheme type box, which identifies the protection scheme used to protect the media, and the Scheme information box, which contains scheme-specific data (defined for each scheme). It is out of the scope of this specification to define a protection scheme.

The Protection scheme information box and its contained boxes are defined in Tables 10.5 – 10.8.

Table 10.5: ProtectionSchemeInfoBox

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'sinf'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
OriginalFormatBox		Box containing identifying the original format	
SchemeTypeBox		Box containing the protection scheme.	
SchemeInformationBox		Box containing the scheme information.	

Table 10.6: OriginalFormatBox

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'frma'
DataFormat	Unsigned int(32)	original format	

DataFormat identifies the format (sample entry) of the decrypted, encoded data. The currently defined formats in 3GP files include 'mp4v', 'h263', 'mp4a', 'samr', 'sawb' and 'tx3g'.

Table 10.7: SchemeTypeBox

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'schm'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0 or 1
SchemeType	Unsigned int(32)	four-character code identifying the scheme	
SchemeVersion	Unsigned int(16)	Version number	
SchemeURI	Unsigned int(8)[ ]	Browser URI (null-terminated UTF-8 string). Present if (Flags & 1) true	

SchemeType and **SchemeVersion** identifiy the encryption scheme and its version. As an option, it is possible to include **SchemeURI** with an URI pointing to a web page for users that don't have the encryption scheme installed.

Table 10.8: SchemeInformationBox

Field	Type	Details	Value
BoxHeader.Size	Unsigned int(32)		
BoxHeader.Type	Unsigned int(32)		'schi'
BoxHeader.Version	Unsigned int(8)		0
BoxHeader.Flags	Bit(24)		0
		Box(es) specific to scheme identified by SchemeType	

The boxes contained in the Scheme information box are defined by the scheme type, which is out of the scope of this specification to define.

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2004-03	*23*	SP-040065			*Presentation to TSG#23 for approval*	1.*0.0*	*2.0.0*