3GPP TSG SA WG3 Security — S3#22 **S3-020019**

25 - 28 February 2002

Bristol, UK

3GPP TSG SA WG4 Meeting #19~~8~~ ***Tdoc* S4-010660**
Tokyo, Japan, 3rd – 7th December 2001

Title:	**Reply to "Liaison Statement on Extended Streaming Service"**
Source:	**TSG SA4**
To:	**TSG T2**
Cc:	**TSG SA1, TSG SA3**
Response to:	Liaison Statement on Extended Streaming Service sent from TSG-T2#14

Contact Person

Name:	Olle Franceschi
Tel. Number:	+46 46 232664
E-mail Address:	Olle.Franceschi@emp.ericsson.se

Attachments: Latest drafts of TS 26.233 Rel-5 and TS 26.234 Rel-5

SA4 wishes to thank T2 for its interest in the PSS Rel-5. Especially T2 requested information regarding status of the following:

- Upload streaming and
- Digital rights management (DRM).

Regarding upload streaming SA4 will probably not be able to finalise the work in time for Rel-5, i.e. this feature will most likely not be a part of PSS Rel-5.

Regarding DRM, T2 might already have been notified that there has started a new WI targeted for Rel-6 in SA1. The work on the service requirements for DRM is ongoing in SA1 at the moment. SA4 has been informed by SA1 that no service requirement for DRM will be included in the Stage 1 specification TS 22.233 for PSS Rel-5. Therefore SA4 is at the moment not conducting any work on DRM in the Rel-5 time frame. Instead SA4 is awaiting the service requirements for DRM from SA1 and will start this work as soon as SA1 has finalised its work on the Stage 1 for DRM.

As requested by T2 the latest draft of the TS 26.233 Rel-5 and TS 26.234 Rel-5 are attached to this LS. T2 also requested SA4 to send a representative to the next T2 meeting to present extended PSS. If T2 still thinks this is needed the rapporteur of the extended PSS WI (Mr. Olle Franceschi) is available on request. Please contact Mr. Olle Franceschi directly (see contact information above).

3GPP TS 26.233 V0.3.0 (2001-12)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Transparent end-to-end packet switched streaming service (PSS);
General description
(Release 5)
TSG-SA4 PSM SWG internal working draft

Keywords

Streaming, multimedia, codec, protocols, packet-switched

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

Contents

Foreword

The 3rd Generation Partnership Project (3GPP) Technical Specification Group (TSG) Services and Systems Aspects has produced this Technical Specification (TS).

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 specification;

The 3GPP packet-switched streaming service (PSS) specification consists of ~~two~~ three 3G TSs; 3GPP TS 22.233 "Transparent end-to-end packet switched streaming service; Stage 1 [6], 3GPP TS 26.234 "Transparent end-to-end packet switched streaming service (PSS); Protocols and codecs" [1] and the present document. The service requirements for PSS are listed in [6]. The present document provides an overview of the 3GPP PSS and [1] specifies the set of PSS protocols and codecs used by the service.

Introduction

Streaming refers to the ability of an application to play synchronised media streams like audio and video streams in a continuous way while those streams are being transmitted to the client over a data network.

Applications, which can be built on top of streaming services, can be classified into on-demand and live information delivery applications. Examples of the first category are music and news-on-demand applications. Live delivery of radio and television programs are examples of the second category.

Streaming over fixed-IP networks is already a major application today. While IETF and W3C have developed a set of protocols used in fixed-IP streaming services, no complete standardised streaming framework has yet been defined. For 3G systems, the 3G packet-switched streaming service (PSS) fills the gap between 3G MMS, e.g. downloading, and conversational services.

PSS enables mobile streaming applications, where the protocol and terminal complexity is lower than for conversational services, which in contrast to a streaming terminal require media input devices, media encoders and more complex protocols.

This document describes the transparent 3G packet-switched streaming services (3G PSS) on a general application level.

1 Scope

The present document contains a general description of a transparent packet-switched streaming service in 3G networks. In particular, it defines the usage scenarios, overall high level end-to-end service concept, and lists terminal related functional components. It also lists any identified service interworking requirements. PSS protocols and codecs are defined in [1].

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.

This specification may contain references to pre-Release-4 GSM specifications. These references shall be taken to refer to the Release 5~~Release 4~~ version where that version exists. Conversion from the pre-Release-4 number to the Release 4 (onwards) number is given in subclause 6.1 of 3GPP TR 41.001[2].

Editor's Note: the Release texts in above paragraph should be checked against latest MCC guideline before creating final Rel 5 version.

[1] 3GPP TS 26.234:"Transpatent end-to-end packet switched streaming service, Protocols and codecs".

[2] 3GPP TR 41.001: "GSM Release specifications".

[3] 3GPP TS 22.140: " Multimedia Messaging Service".

[4] 3GPP TS 23.140: "Multimedia Messaging Service (MMS); Functional description; Stage 2".

[5] 3GPP TS 23.060: "General Packet Radio Service (GPRS); Service description; Stage 2".

[6] 3GPP TS 22.233: "Transparent end-to-end packet switched streaming service, Service aspects; Stage 1".

3 Abbreviations

For the purposes of the present document, the following abbreviations apply

DRM	Digital Rights Management
GIF	Graphics Interchange Format
HTML	HyperText Markup Language
IETF	Internet Engineering Task Force
IP	Internet Protocol
MMS	Multimedia Messaging Service
PDP	Packet Data Protocol
PSS	Packet-switched Streaming Service
RAB	Radio Access Bearer
RFC	IETF Request For Comments
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SDP	Session Description Protocol

TCP	Transport Control Protocol
UDP	User Datagram Protocol
URI	Universal Resource Identifier
WAP	Wireless Application Protocol
WWW	World Wide Web

4 Usage scenarios

4.1 Applications

The streaming platform supports a multitude of different applications including streaming of news at very low bitrates using still images and speech, music listening at various bitrates and qualities, video clips and watching live sports events.

4.2 Use case descriptions

4.2.1 Simple streaming

The simple streaming service includes a basic set of streaming control protocols, transport protocols, media codecs and scene description protocol. In this ~~simple~~ basic case defined first time in Release 4 version of this specification, there is neither explicit capability exchange, nor any encryption or digital rights management.

A mobile user gets a URI to specific content that suits his or her terminal. This URI may come from a WWW-browser, a WAP-browser, or typed in by hand. This URI specifies a streaming server and the address of the content on that server. A PSS~~n~~ application that establishes the multimedia session shall understand a Session Description Protocol (SDP) file. Sessions containing only non-streamable content such as a SMIL file, still images and text to form a time-synchronised presentation don't require use of SDP file in session establishment. Instead HTTP protocol shall be used for receiving the presentation files. PSS SMIL sessions can also include URIs to streamable content, requiring parsing a SDP file and/or RTSP signalling.

The SDP file may be obtained in a number of ways. It may be provided in a link inside the HTML page that the user downloads, via an embed tag. It may also be directly obtained by typing it as a URI. It may also be obtained through RTSP signalling via the DESCRIBE method. The SDP file contains the description of the session (session name, author, …), the type of media to be presented, and the bitrate of the media.

The session establishment is the process in which the browser or the mobile user invokes a streaming client to set up the session against the server. The UE is expected to have an active PDP context in accordance with [5] or other type of radio bearer that enables IP packet transmission at the start of session establishment signalling. The client may be able to ask for more information about the content. The client shall initiate the provisioning of a bearer with appropriate QoS for the streaming media. ~~Sessions containing only non-streamable content such as a SMIL file, still images and text to form a time-synchronised presentation don't require use of SDP file in session establishment.~~ The set up of the streaming service is done by sending an RTSP SETUP message for each media stream chosen by the client. This returns the UDP and/or TCP port etc. to be used for the respective media stream. The client sends a RTSP PLAY message to the server that starts to send one or more streams over the IP network.

This case is illustrated below in figure 1.

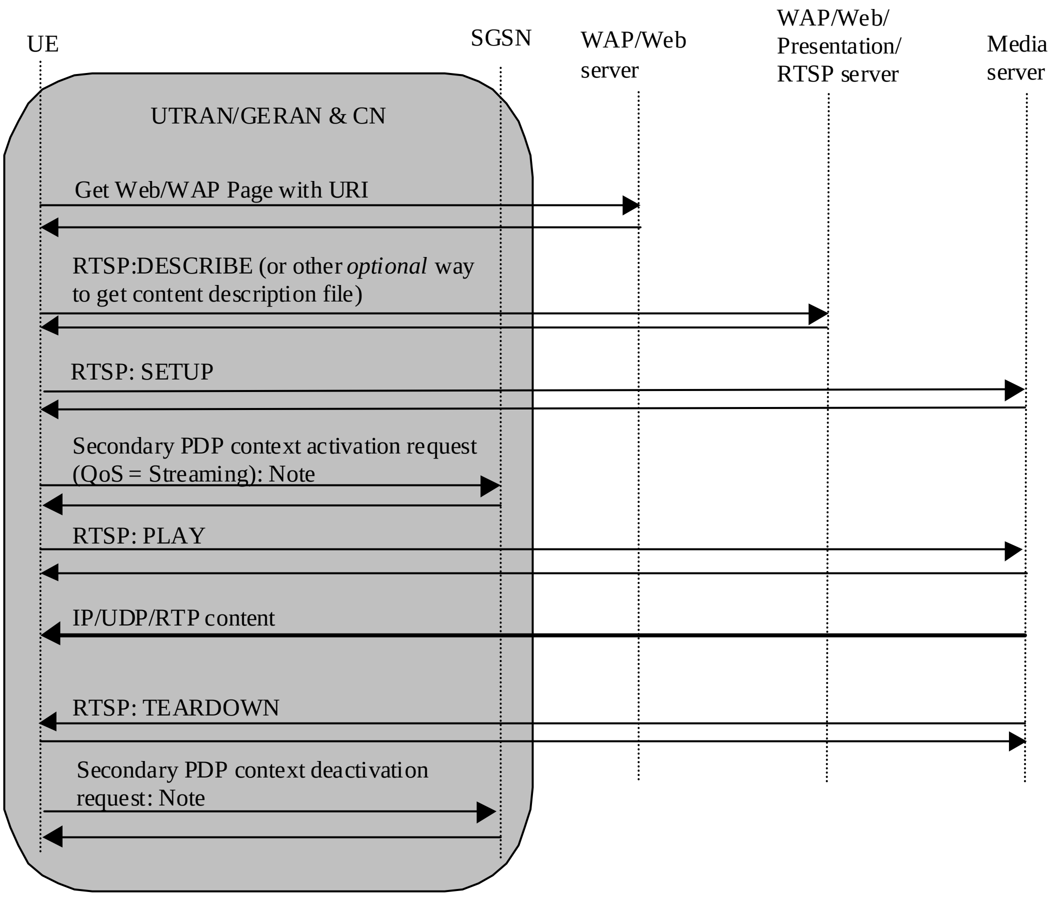

UE
SGSN
WAP/Web server
WAP/Web/ Presentation/ RTSP server
Media server
UTRAN/GERAN & CN
Get Web/WAP Page with URI
RTSP:DESCRIBE (or other *optional* way to get content description file)
RTSP: SETUP
Secondary PDP context activation request (QoS = Streaming): Note
RTSP: PLAY
IP/UDP/RTP content
RTSP: TEARDOWN
Secondary PDP context deactivation request: Note

Figure 1: Schematic view of a simple streaming session

[Editor's note: the arrow directions in RTSP:TEARDOWN need to be changed to opposite (in final CR created for Release 5)].

Note: These messages are one example of how to establish and terminate the bearer with the desired QoS. Other alternatives exist, e.g., an existing PDP context might be modified.

4.2.2 Other streaming cases

~~Extended~~ The streaming service defined in [1] will support all features defined for the ~~simple~~ Release 4 streaming case in a fully backwards compatible manner, and may additionally include more advanced service features, such as capability exchange, interworking with core network services~~,~~ and security. ~~and Digital Rights Management. This extended~~More extensions to ~~set of~~ PSS features, such as Digital Rights Management can be specified in future 3GPP Releases.

[Editor's Note: The use case description in this subchapter needs to be reworded regarding the available advanced services for the Release 5 version of 26.233 to reflect the scope of extended streaming decided for the particular release. New terminal functional components - if any - should be collected below clause 6, the descriptions of other features to new clauses after Sec. 10.]

Editor's Note: in early December 2001 only capability exchange of the above section's listed features has been included to the PSS Release 5 into 26.234. The final outcome of Rel 5 content in March 2002 will define which features remain for "future 3GPP releases".

5 General service architecture

Figure 2 shows the most important service specific entities involved in a 3G packet -switched streaming service. A streaming service requires at least a content server and a streaming client. A streaming server is located behind the Gi interface. Additional components like portals, profile servers, caching servers and proxies located behind the Gi interface might be involved as well to provide additional services or to improve the overall service quality.

Portals are servers allowing convenient access to streamed media content. For instance, a portal might offer content browse and search facilities. In the simplest case, it is simply a Web/WAP-page with a list of links to streaming content. The content itself is usually stored on content servers, which can be located elsewhere in the network.

User and ~~terminal~~ device profile servers are used to store user preferences and ~~terminal~~ device capabilities. This information can be used to control the presentation of streamed media content to a mobile user. A high-level illustration of the capability exchange framework can be seen in figure 3.

NOTE: specific user preference attributes have not yet been defined for PSS. The extensible device capability attributes allow specifying such attributes in the future releases.

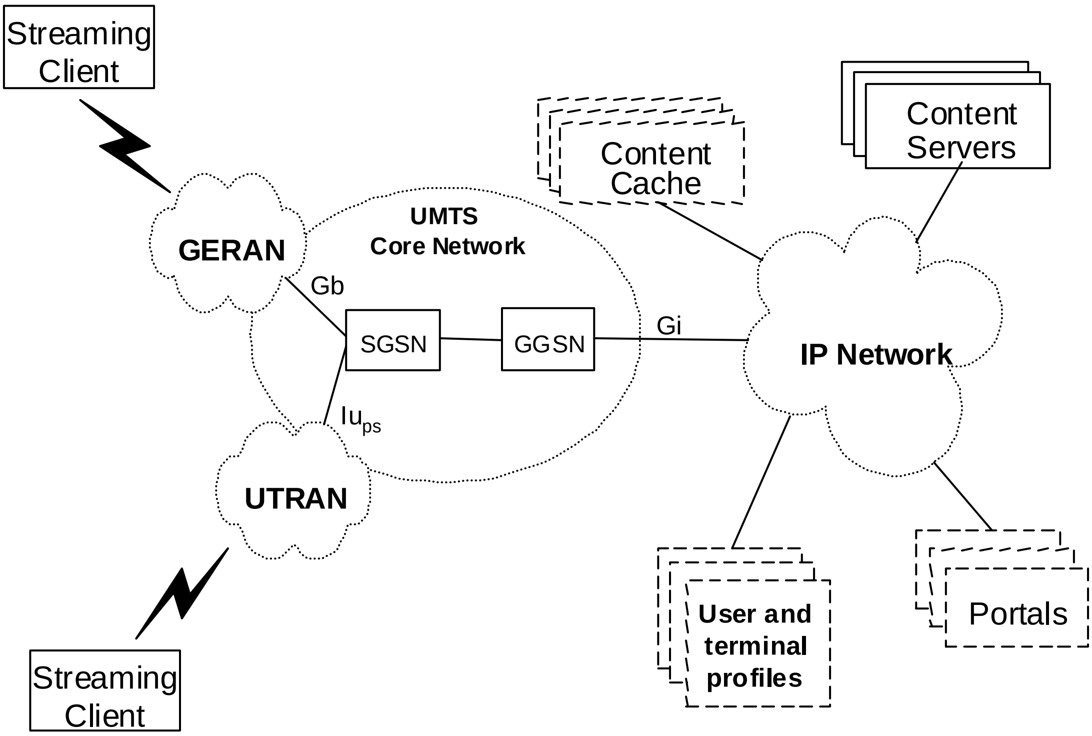


Figure 2: Network elements involved in a 3G packet switched streaming service

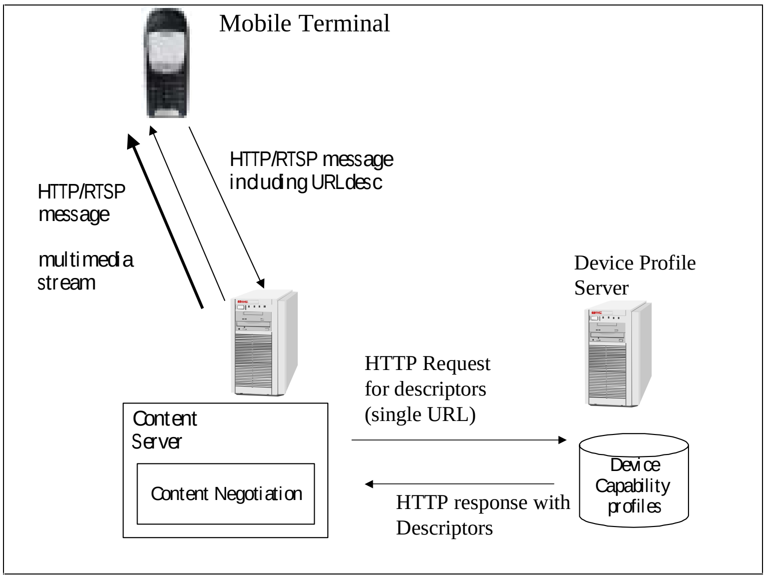

Mobile Terminal
HTTP/RTSP message
multimedia stream
HTTP/RTSP message including URLdesc
Content Server
Content Negotiation
HTTP Request for descriptors (single URL)
HTTP response with Descriptors
Device Profile Server
Device Capability profiles

Figure 3. Logical system architecture of the capability negotiation mechanism applied in PSS.

6 Functional components of a PSS terminal

This chapter lists the 3G packet-switched streaming service components, which belong to the terminal. Note that not all of the components need to be mandatory. The functional behaviour of the different components is discussed in the following.

6.1 Session protocols and data transport

Protocols are needed for PSS session establishment, capability exchange, session setup, session control, scene description, and data transport of streaming media and other data. The PSS protocols to be used are specified in [1].

Note that for the simple streaming case defined in subclause 4.2.1, no specific capability exchange protocol in addition to the session description mechanism is required. ~~More complex streaming services with detailed capability exchange mechanism can be specified in future 3GPP Releases.~~

~~In extended streaming service, t~~The normative part of device capability and user preference profile exchange mechanism is defined in clause 5.2 and Annex E of [1]. An informative part is included in Annex A.3 of [1].

6.2 Codecs

Codecs are needed for speech, natural and synthetic audio, video, still images, bitmap graphics, vector graphics and text. The codecs to be used are specified in [1]. ~~Vector graphics belongs to the extended PSS features and is expected to be specified in future 3GPP Release.~~

[Editor's Note: any totally new codec types not present in current text should be reflected in this clause. For example: sentence on vector graphics should be removed if it is agreed for the Release 5 PSS codec set.]

7 File format

The file format is an important element of the content manipulation chain. Conceptually, there is a difference between the coding format and the file format. The coding format is related to the action of a specific coding algorithm that codes the content information into a codestream. The file format is instead a way of organising the prestored codestream in such way that it can be accessed for local decoding and playback, or transferred as a file on different media, or streamed over different transport. Some file formats are optimised for one or more of these functions, others aim instead at achieving a higher flexibility.

When a single media type is involved, the coding and the file format are often considered, and referred to, as a single entity. When multimedia information is involved, instead, it is appropriate to maintain, at least conceptually, the distinction between these two instances. The file format can play an important role in facilitating the organisation and the access to the coded information, independently of the specific coding formats.

The clause 9 in [1] specifies how the 3GPP MMS [3] shall utilise a file format. The format is specified in order to enable standardised content transport chain that enables downlink streaming of MMS messages. See also clause 8.2.

8 Interworking with other core network services

8.1 Interworking with WAP

Not required.

8.2 Interworking with MMS

TS 23.140 [4] defines a new optional feature for the MMS, which enables streaming of the MMS messages by the message recipient. The MMS streaming option uses the codecs and protocols in accordance with TS 26.234.

Additionally, [4] mandates the use of the interchange format recommendation specified in 26.234, clause 9 for MMS purposes.

[Editor's Note: any approved new general aspects to MMS interworking brougt by Release 5 (file download and/or up-streaming) should be reflected in this clause.]

8.3 Interworking with charging/billing services

Interworking with charging/billing services can be part of the ~~extended~~ PSS. Detailed Call Detailed Record structures and charging mechanisms applied for PSS are specified in TS xx.yyy [z] clause a.b.

[Editor's note: SA4 is not working on charging aspects, and the outcome of work done by SA5 for PSS charging will be referred above when available.]

9 Security

Streaming security mechanisms can be part of the ~~extended~~ PSS.

[Editor's note: secure streaming is a listed requirement in TS 22.233 - if a technical solution is agreed for Release 5 in appropriate technical working group (SA3 & SA4 jointly?), corresponding information will be referenced to and explained here briefly. Otherwise this item may move forward to Release 6.]

10 Digital Rights Management

Standardisation of 3G PSS needs to be aligned with standardised or industry solutions for media rights management. An appropriate DRM framework has been identified as a new active 3GPP feature to be completed for next 3GPP release.~~can be part of the extended PSS.~~

[Editor's note: SA1's recent decision in SA1#14 was to leave DRM requirement for PSS for Release 6, where a common 3GPP work item on DRM is due to be finished.]

Annex A (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New

Change history for TSG-SA4 PSM SWG internal working draft

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2001-06-07				-	First draft for Release 5 during SA4#17.	TS26233-400	S4-010423
2001-08-29				0.0.1	Editorial changes before SA4#18	S4-010423	S4-010475
2001-09-05				0.1.0	Changes resulting from SA4#18	S4-010475	S4-010523
2001-12-02				0.2.0	Update to SA4#19 - removed old simple/extended PSS terminology in 4.2.1 & 4.2.2 and elsewhere in text, modified clause 6.1, 8.3, 9. Terminology of 26.234 on device capability profiles applied.	S4-010523	S4-010588R
2001-12-03				0.3.0	Above + change to Figure 1 + removal of DRM from PSS Release 5 in 4.2.2	S4-010588R	S4-010649

3GPP TS 26.234 V0.5.1 (2001-12-07)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Transparent end-to-end packet switched streaming service (PSS);
Protocols and codecs
(Release 5)
TSG-SA4 PSM SWG internal working draft

Editor's Note: Information for the reader

The present document is the TSG-SA4 PSM SWG internal working draft of TS 26.234 Release 5. The intention is that the text highlighted with grey in the present document will be included in one or several Change Requests (CR) that will be sent to SA for approval and become Release 5 of TS 26.234. During the development of Release 5 (this document) it is anticipated that changes will also be done to the release 4 version of TS 26.234. Note that any changes to Release 4 have to be written as formal CRs (see http://www.3gpp.org/3G_Specs/CRs.htm) using the appropriate CR cover sheet. When such changes/correction to Release 4 have been approved by TSG-SA they will be included in the present document. They will however neither be marked with grey nor with MS Word change marks since they belong to the Release 4 version. The following apply in the present document:

- Text highlighted with grey indicates additions compared to Release 4 that will be included in Release 5. Text marked with ~~grey~~ indicates Release 4 text that will be deleted in the Release 5 version.
- MS Word change marks will be used to indicate changes between successive version of the draft (e.g.. the difference between version 0.0.1 and 0.0.2).
- Editor's notes are just for information during the drafting. The may e.g. be used to describe the current status of the work in the PSM subgroup.
- Editor's Notes will not be part of the final Release 5 of TS 26.234 and are therefore not highlighted with grey.
- Changes to figures in the Release 5 version are indicated by an Editor's note below the figure.
- Text highlighted with yellow is used for giving important information to the reader. Yellow text will not be part of the final Release 5 document.

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organisational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organisational Partners accept no liability for any use of this Specification.
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organisational Partners' Publications Offices.

Keywords

UMTS, IP, packet mode, protocol, codec

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

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 specification;

The 3GPP transparent end-to-end packet-switched streaming service (PSS) specification consists of three~~two~~ 3G TSs; 3GPP TS 22.233 [1], 3GPP TS 26.233 [2] and the present document. The first TS contains the service requirements for the PSS, the second provides an overview of the 3GPP PSS and the present document the details of protocol and codecs used by the service.

Introduction

Streaming refers to the ability of an application to play synchronised media streams like audio and video streams in a continuous way while those streams are being transmitted to the client over a data network.

Applications, which can be built on top of streaming services, can be classified into on-demand and live information delivery applications. Examples of the first category are music and news-on-demand applications. Live delivery of radio and television programs are examples of the second category.

The 3GPP PSS provides a framework for Internet Protocol (IP) based streaming applications in 3G networks.

1 Scope

The present document specifies the protocols and codecs for the PSS within the 3GPP system. Protocols for control signalling, capability exchange, scene description, media transport and media encapsulations are specified. Codecs for speech, natural and synthetic audio, video, still images, bitmap graphics, vector graphics and text are specified.

The present document is applicable to IP based packet switched networks.

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: "Service aspects; Stage 1 Transparent End-to-End Packet-switched Streaming Service".~~(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: 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".

[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): "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] IETF RFC 2083: "PNG (Portable Networks Graphics) Specification version 1.0 ", T. Boutell, et. al., March 1997.

[37] W3C Working Draft Recommendation: "CC/PP structure and vocabularies", http://www.w3.org/Mobile/CCPP/Group/Drafts/WD-CCPP-struct-vocab-20010620/, June 2001.

[38] WAP UAProf Specification, http://www1.wapforum.org/tech/documents/WAP-248-UAProf-20011009-p.pdf, May 2001.

[39] W3C Candidate Recommendation: "Resource Description Framework (RDF) Schema Specification 1.0", http://www.w3.org/TR/2000/CR-rdf-schema-20000327, March 2000.

[40] W3C Recommendation: "Scalable Vector Graphics (SVG) 1.0 Specification", http://www.w3.org/TR/SVG/, September 2001.

[41] Scalable Polyphony MIDI Specification, MIDI Manufacturers Association (2001)

[42] Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP, MIDI Manufacturers Association (2001).

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 and video

device capability description: a description of device capabilities and/or user preferences. Contains a number of capability attributes.

device capability profile: same as device capability description

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

presentation description: contains information about one or more media streams within a presentation, such as the set of encodings, network addresses and information about the content

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

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

scene description: description of the spatial layout and temporal behaviour of a presentation, it can also contain hyperlinks

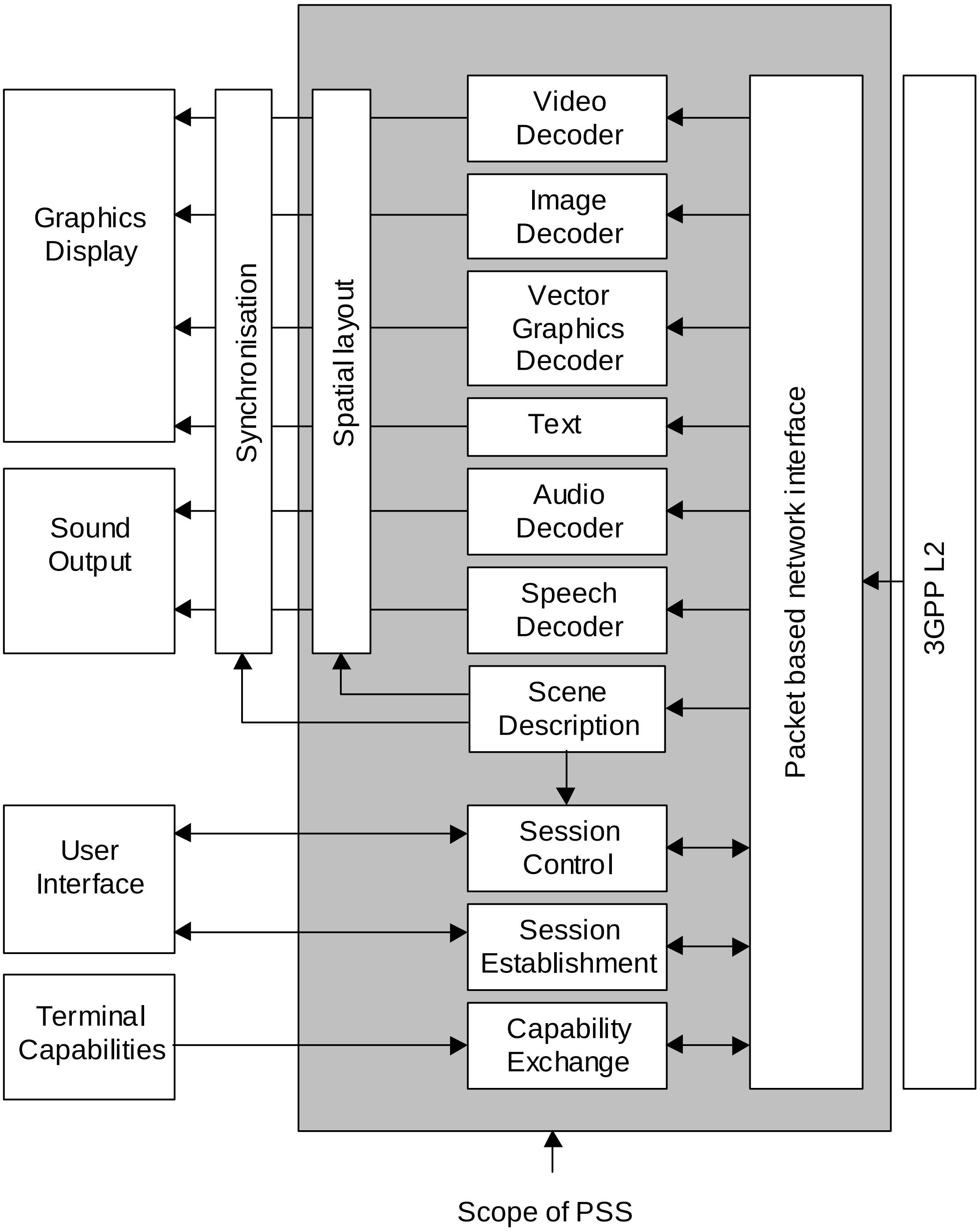
3.2 Abbreviations

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

AAC	Advanced Audio Coding
BIFS	Binary Format for Scene description
CC/PP	Composite Capability / Preference Profiles
DCT	Discrete Cosine Transform
GIF	Graphics Interchange Format
HTML	Hyper Text Markup Language
ITU-T	International Telecommunications Union – Telecommunications
JFIF	JPEG File Interchange Format
MIME	Multipurpose Internet Mail Extensions
MMS	Multimedia Messaging Service
MP4	MPEG-4 file format
PSS	Packet-switched Streaming Service
QCIF	Quarter Common Intermediate Format
RDF	Resource Description Framework
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SDP	Session Description Protocol
SMIL	Synchronised Multimedia Integration Language

SP-MIDI	Scalable Polyphony Musical Instrument Digital Interface
UAProf	User Agent Profile
UCS-2	Universal Character Set (the two octet form)
UTF-8	Unicode Transformation Format (the 8-bit form)
W3C	WWW Consortium
WML	Wireless Markup Language
XHTML	eXtensible Hyper Text Markup Language
XML	eXtensible Markup Language

4 System description



~~NOTE: Dashed components are not specified for the simple PSS.~~

Figure 1: Functional components of a PSS client

Editor's Note: Figure 1 has been modified and it need to be included in the CR for Release 5.

Editor's Note: Synthetic audio needs to be reflected in clause 4.

Figure 1 shows the functional components of a PSS client. Figure 2 gives an overview of the protocol stack used in a PSS client and also shows a more detailed view of the packet based network interface. The functional components can be divided into control, scene description, media codecs and the transport of media and control data. ~~TS 26.233 [2] defines the simple and extended PSS. Dashed functional components in figure 1 are not specified for the simple PSS.~~

The control related elements are session establishment, capability exchange and session control (see clause 5).

- Session establishment refers to methods to invoke a PSS session from a browser or directly by entering an URL in the terminal's user interface.
- Capability exchange enables choice or adaptation of media streams depending on different terminal capabilities.
- Session control deals with the set-up of the individual media streams between a PSS client and one or several PSS servers. It also enables control of the individual media streams by the user. It may involve VCR-like presentation control functions like start, pause, fast forward and stop of a media presentation.

The scene description consists of spatial layout and a description of the temporal relation between different media that is included in the media presentation. The first gives the layout of different media components on the screen and the latter controls the synchronisation of the different media (see clause 8).

The PSS includes media codecs for video, still images, vector graphics, bitmap graphics, text, natural and synthetic audio, and speech (see clause 7).

Transport of media and control data consists of the encapsulation of the coded media and control data in a transport protocol (see clause 6). This is shown in figure 1 as the "packet based network interface" and displayed in more detail in the protocol stack of figure 2.

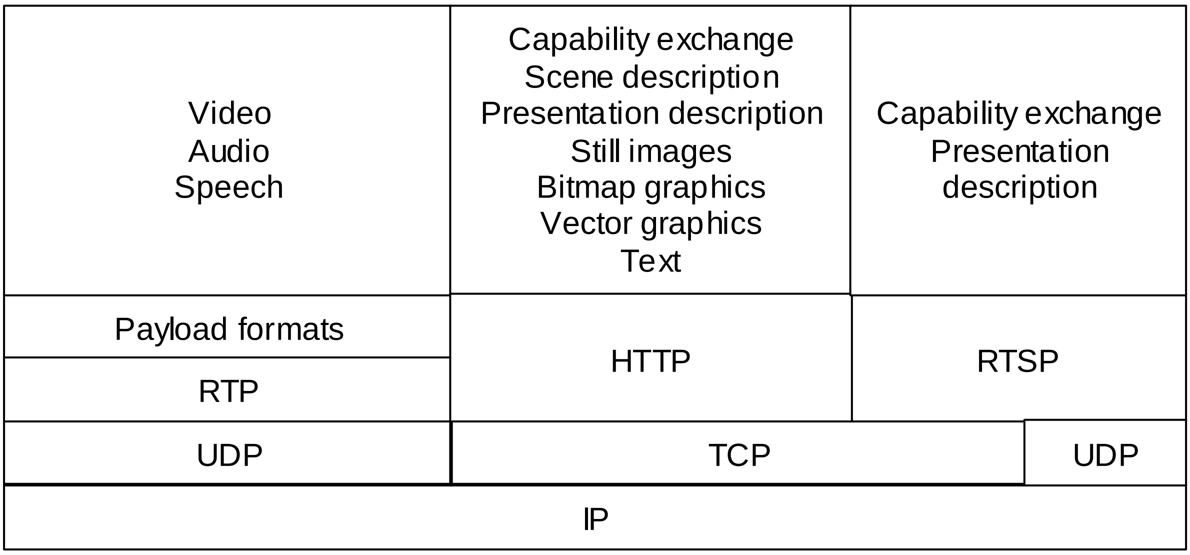


Figure 2: Overview of the protocol stack

Editor's Note: Figure 2 has been modified and it needs to be included in the CR for Release 5.

5 Protocols

5.1 Session establishment

Session establishment refers to the method by which a PSS client obtains the initial session description. The initial session description can e.g. be a presentation description, a scene description or just an URL to the content.

A PSS client shall support initial session descriptions specified in one of the following formats: SMIL, SDP, or plain RTSP URL.

In addition to rtsp:// the PSS client shall support URLs [4] to valid initial session descriptions starting with file:// (for locally stored files) and http:// (for presentation descriptions or scene descriptions delivered via HTTP).

Examples for valid inputs to a PSS client are: file://temp/morning_news.smil, http://mediaportal/morning_news.sdp, and rtsp://mediaportal/morning_news.

URLs can be made available to a PSS client in many different ways. It is out of the scope of this recommendation to mandate any specific mechanism. However, an application using the 3GPP PSS shall at least support URLs of the above type, specified or selected by the user.

The preferred way would be to embed URLs to initial session descriptions within HTML or WML pages. Browser applications that support the HTTP protocol could then download the initial session description and pass the content to the PSS client for further processing. How exactly this is done is an implementation specific issue and out of the scope of this recommendation.

5.2 Capability exchange

5.2.1 General

Capability exchange is an important functionality in the PSS. It enables PSS servers to provide a wide range of devices with content suitable for the particular device in question. Another very important task is to provide a smooth transition between different releases of PSS. Therefore, PSS clients and servers should support capability exchange.

The specification of capability exchange for PSS is divided into two parts. The normative part contained in clause 5.2 and an informative part in clause A.3 in Annex A of the present document. The normative part gives all the necessary requirements that a client or server shall conform to when implementing capability exchange in the PSS. The informative part provides additional important information for understanding the concept and usage of the functionality. It is recommended to read clause A.3 in Annex A before continuing with clauses 5.2.2-5.2.7.

5.2.2 The device capability profile structure

A device capability profile is a RDF [39] document that follows the structure of the CC/PP framework [37] and the CC/PP application UAProf [38]. Attributes are used to specify device capabilities and preferences. A set of attribute names, permissible values and semantics constitute a CC/PP vocabulary, which is defined by a RDF schema. For PSS the UAProf vocabulary is reused and an additional PSS specific vocabulary is defined. The details can be found in clause 5.2.3. The syntax of the attributes is defined in the vocabulary schema but also, to some extent, the semantics. A PSS device capability profile is an instance of the schema (UAProf and/or the PSS specific schema) and shall follow the rules govern the formation of a profile given in the CC/PP specification [37]. The profile schema shall also be governed by the rules defined in UAProf [38] chapter 7, 7.1, 7.3 and 7.4.

5.2.3 Vocabularies for PSS

5.2.3.1 General

Clause 5.2.3 specifies the attribute vocabularies to be used by the PSS capability exchange.

PSS servers should understand the attributes in both the streaming component of the PSS base vocabulary and the recommended attributes from the UAProf vocabulary [38]. It may additional support other UAProf attributes.

5.2.3.2 PSS base vocabulary

The PSS base vocabulary contains one component called "Streaming". A vocabulary extension to UAProf shall be defined as a RDF schema. This schema can be found in Annex E. The schema together with the description of the attributes in the present clause, defines the vocabulary. The vocabulary is associated with an XML namespace, which combines a base URI with a local XML element name to yield a URI. Annex E provides the details.

All PSS attributes are put in a PSS specific component called “Streaming”. The list of PSS attributes is as follows:

Attribute name: **AudioChannels**

Attribute definition: This attribute defines the audio output channels.

Component: Streaming

Type: Literal

Legal values: “Mono”, “Stereo”

Resolution rule: Locked

EXAMPLE 1: `<AudioChannels>Mono</AudioChannels>`

Attribute name: **VideoPreDecoderBufferSize**

Attribute definition: This attribute signals if the optional video buffering requirements defined in Annex F are supported. It also defines the size of the hypothetical pre-decoder buffer defined in Annex F. A value equal to zero means that Annex F is not supported. A value equal to one means that Annex F is supported. In this case the size of the buffer is the default size defined in Annex F. A value equal to or greater than the default buffer size defined in Annex F means that Annex F is supported and sets the buffer size to the given number of octets.

Component: Streaming

Type: Number

Legal values: Integer value equal to or greater than zero. Values greater than one but less than the default buffer size defined in Annex F are not allowed.

Resolution rule: Locked

EXAMPLE 2: `<VideoPreDecoderBufferSize>30720</VideoPreDecoderBufferSize>`

Attribute name: **VideoInitialPostDecoderBufferingPeriod**

Attribute definition: If Annex F is not supported, the attribute has no meaning. If Annex F is supported, this attribute defines the maximum initial post-decoder buffering period of video. Values are interpreted as clock ticks of a 90-kHz clock. In other words, the value is incremented by one for each 1/90 000 seconds. For example, the value 9000 corresponds to 1/10 of a second initial post-decoder buffering.

Component: Streaming

Type: Number

Legal values: Integer value equal to or greater than zero.

Resolution rule: Locked

EXAMPLE 3:
```
<VideoInitialPostDecoderBufferingPeriod>9000
</VideoInitialPostDecoderBufferingPeriod>
```

Attribute name: **PssAccept**

Attribute definition: List of content types (MIME types) the PSS application supports. Both CcppAccept (SoftwarePlatform, UAProf) and PssAccept can be used but if PssAccept is defined it has precedence over CcppAccept and a PSS application shall then use PssAccept.

Component: Streaming

Type: Literal (Bag)

Legal values: List of MIME types with related parameters.

Resolution rule: Append

EXAMPLE 4:
```
<PssAccept>
  <rdf:Bag>
    <rdf:li>audio/AMR-WB; octet-alignment</rdf:li>
    <rdf:li>application/smil</rdf:li>
  </rdf:Bag>
</PssAccept>
```

Attribute name: **PssAccept-Subset**

Attribute definition: List of content types for which the PSS application supports a subset. MIME-types can in most cases effectively be used to express variations in support for different media types. Many MIME-types, e.g. AMR-NB has several parameters that can be used for this purpose. There may exist content types for which the PSS application only supports a subset and this subset can not be expressed with MIME-type parameters. In these cases the attribute PssAccept-Subset is used to describe support for a subset of a specific content type. If a subset of a specific content type is declared in PssAccept-Subset, this means that PssAccept-Subset has precedence over both PssAccept and CcppAccept if the same content-type is listed there.

This is illustrated with an example. If PssAccept="audio/AMR", "image/jpeg" and PssAccept-Subset="JPEG Baseline" then "audio/AMR" and JPEG Base line is supported. "image/jpeg" in PssAccept is of no importance since it’s related to "JPEG Baseline" in PssAccept-Subset. Subset identifiers and corresponding semantics shall only be defined by the TSG responsible for the present document..

Component: Streaming

Type: Literal

Legal values: "JPEG Baseline"

Resolution rule: Append

EXAMPLE 5: `<PssAccept-Subset>JPEG Baseline</PssAccept-Subset>`

Attribute name: **PssVersion**

Attribute definition: PSS version supported by the client.

Component: Streaming

Type: Literal

Legal values: "3GPP-R4", "3GPP-R5" and so forth.

Resolution rule: Locked

EXAMPLE 6: `<PssVersion>3GPP-R4</PssVersion>`

Attribute name: **RenderingScreenSize**

Attribute definition: The rendering size of the device’s screen in unit of pixels. The horizontal size is given followed by the vertical size.

Component: Streaming

Type: Dimension

Legal values: Two integer values equal or greater than zero. A value equal “0x0”means that there exist no possibility to render visual PSS presentations.

Resolution rule: Locked

EXAMPLE 7: `<RenderingScreenSize>70x15</RenderingScreenSize>`

Attribute name: **SmilBaseSet**

Attribute definition: Indicates a base set of SMIL 2.0 modules that the client supports.

Component: Streaming

Type: Literal

Legal values: Pre-defined identifiers. “SMIL-3GPP-R4" includes all SMIL 2.0 modules required for scene description support according to clause 8 of the present document.

Resolution rule: Locked

EXAMPLE 8: `<SmilBaseSet>SMIL-3GPP-R4</SmilBaseSet>`

Attribute name: **SmilModules**

Attribute definition: This attribute defines a list of SMIL 2.0 modules supported by the client. If the SmilBaseSet is used those modules do not need to be explicitly listed here. In that case only additional module support needs to be listed.

Component: Streaming

Type: Literal (Bag)

Legal values: SMIL 2.0 module names defined in the SMIL 2.0 recommendation [31], section 2.3.3, table 2.

Resolution rule: Append

EXAMPLE 9:

```
<SmilModules>
  <rdf:Bag>
    <rdf:li>BasicTransitions</rdf:li>
    <rdf:li>MulitArcTiming</rdf:li>
  </rdf:Bag>
</SmilModules>
```

5.2.3.3 Attributes from UAProf

In the UAProf vocabulary [38] there are several attributes that are of interest for the PSS. The formal definition of these attributes is given in [38]. The following list of attributes is recommended for PSS applications:

Attribute name: **BitsPerPixel**

Component: HardwarePlatform

Attribute description: The number of bits of colour or greyscale information per pixel

EXAMPLE 1: `<BitsPerPixel>8</BitsPerPixel>`

Attribute name: **ColorCapable**

Component: HardwarePlatform

Attribute description: Whether the device display supports colour or not.

EXAMPLE 2: `<ColorCapable>Yes</ColorCapable>`

Attribute name: **PixelAspectRatio**

Component: HardwarePlatform

Attribute description: Ratio of pixel width to pixel height

EXAMPLE 3: `<PixelAspectRatio>1x2</PixelAspectRatio>`

Attribute name: **PointingResolution**

Component: HardwarePlatform

Attribute description: Type of resolution of the pointing accessory supported by the device.

EXAMPLE 4: `<PointingResolution>Pixel</PointingResolution>`

Attribute name: **SoundOutputCapable**

Component: HardwarePlatform

Attribute description: Indicates whether the device supports sound output

EXAMPLE 5: `<SoundOutputCapable>Yes</SoundOutputCapable>`

Attribute name: **Model**

Component: HardwarePlatform

Attribute description: Model number assigned to the terminal device by the vendor or manufacturer

EXAMPLE 6: `<Model>Lexus</Model>`

Attribute name: **Vendor**

Component: HardwarePlatform

Attribute description: Name of the vendor manufacturing the terminal device

EXAMPLE 7: `<Vendor>Toyota</Vendor>`

Attribute name: **CcppAccept-Charset**

Component: SoftwarePlatform

Attribute description: List of character sets the device supports

EXAMPLE 8:

```
<CcppAccept-Charset>
  <rdf:Bag>
    <rdf:li>UTF-8</rdf:li>
  </rdf:Bag>
</CcppAccept-Charset>
```

Attribute name: **CcppAccept-Encoding**

Component: SoftwarePlatform

Attribute description: List of transfer encodings the device supports

EXAMPLE 9:

```
<CcppAccept-Encoding>
  <rdf:Bag>
    <rdf:li>base64</rdf:li>
  </rdf:Bag>
</CcppAccept-Encoding>
```

Attribute name: **CcppAccept-Language**

Component: SoftwarePlatform

Attribute description: List of preferred document languages

EXAMPLE 10:

```
<CcppAccept-Language>
  <rdf:Seq>
    <rdf:li>en</rdf:li>
    <rdf:li>se</rdf:li>
  </rdf:Seq>
</CcppAccept-Language>
```

5.2.4 Extensions to the PSS schema/vocabulary

The use of RDF enables an extensibility mechanism for CC/PP-based schemas that addresses the evolution of new types of devices and applications. The PSS profile schema specification is going to provide a base vocabulary but in the future new usage scenarios might have need for expressing new attributes. If the base vocabulary is updated a new unique namespace will be assigned to the updated schema. The base vocabulary shall only be changed by the TSG responsible for the present document. All extensions to the profile schema shall be governed by the rules defined in [38] clause 7.7.

5.2.5 Signalling of profile information between client and server

When a PSS client or server support capability exchange it shall support the profile information transport over both HTTP and RTSP between client and server as defined in clause 9.1 (including its subsections) of the WAP 2.0 UAProf specification [38] with the following additions:

- The "x-wap-profile" and "x-wap-profile-diff" headers may not be present in all HTTP or RTSP request. That is, the requirement to send this header in all requests has been relaxed.
- The defined headers may be applied to both RTSP and HTTP.
- The "x-wap-profile-diff" header is only valid for the current request. The reason is that PSS does not have the WSP session concept of WAP.
- Push is not relevant for the PSS.

The following recommendations are made to how and when profile information should be sent between client and server:

- PSS content servers supporting capability exchange shall be able to receive profile information in all HTTP and RTSP requests.
- The terminal should not send the "wap-profile-diff" header over the air-interface since there is no compression scheme defined.
- RTSP: the client should send profile information in the DESCRIBE message. It may send it in any other request.

If the terminal has some prior knowledge about the file type it is about to retrieve, e.g. file extensions, the following apply:

- HTTP and SDP: when retrieving an SDP with HTTP the client should include profile information in the GET request. This way the HTTP server can deliver an optimised SDP to the client.

- HTTP and SMIL: When retrieving a SMIL file with HTTP the client should include profile information in the GET request. This way the HTTP server can deliver an optimised SMIL presentation to the client. A SMIL presentation can include links to static media. The server should optimise the SMIL file so that links to the referenced static media are adapted to the requesting client. When the "x-wap-profile-warning" indicates that content selection has been applied (201-203) the PSS client should assume that no more capability exchange has to be performed for the static media components. In this case it should not send any profile information when retrieving static media to be included in the SMIL presentation. This will minimise the HTTP header overhead.

5.2.6 Merging device capability profiles

Profiles need to be merged whenever the PSS server receives multiple device capability profiles. Multiple occurrences of attributes and default values make it necessary to resolve the profiles according to a resolution process.

The resolution process is the same as defined in UAProf [38] clause 6.4.1.

- Resolve all indirect references by retrieving URI references contained within the profile.

- Resolve each profile and profile-diff document by first applying attribute values contained in the default URI references and by second applying overriding attribute values contained within the category blocks of that profile or profile-diff.

- Determine the final value of the attributes by applying the resolved attribute values from each profile and profile-diff in order, with the attribute values determined by the resolution rules provided in the schema. Where no resolution rules are provided for a particular attribute in the schema, values provided in profile-diffs are assumed to override values provided in previous profiles or profile-diffs.

When several URLs are defined in the "x-wap-profile" header and there exists any attribute that occurs more than once in these profiles the rule is that the attribute value in the second URL overrides the attribute value from the first URL and so forth. This is what is meant with "Determine the final value of the attributes by applying the resolved attribute values from each profile and profile-diff in order, with…" in the third bullet above. If the profile is completely or partly inaccessible or otherwise corrupted the server should still provide content to the client. The server is responsible for delivering content optimised for the client based on the received profile in a best effort manner.

NOTE: For the reasons explained in Annex A clause A.3.3 the usage of indirect references in profiles (using the CC/PP defaults element) is not recommended.

5.2.7 Profile transfer between the PSS server and the device profile server

The device capability profiles are stored on a device profile server and referenced with URLs. According to the profile resolution process in clause 5.2.6 of the present document, the PSS server ends up with a number of URLs referring to profiles and these shall be retrieved.

- The device profile server shall support HTTP 1.1 for the transfer of device capability profiles to the PSS server.

- If the PSS server supports capability exchange it shall support HTTP 1.1 for transfer of device capability profiles from the device profile server. A URL shall be used to identify a device capability profile.

- Normal content caching provisions as defined by HTTP apply.

~~No explicit capability exchange protocol is specified for the simple PSS.. Instead it is assumed that the user is aware of that the content he/she is about to stream fits the capabilities, e.g. screen size, of the particular device used. Protocols for capability exchange can be specified for the extended PSS.~~

5.3 Session set-up and control

5.3.1 General

Continuous media is media that have an intrinsic time line. Discrete media on the other does not it self contain an element of time. In this specification speech, audio and video belongs to first category and still images and text to the latter one. Bitmap graphics can fall into both groups, but is in this specification defined to be discrete media.

Streaming of continuous media using RTP/UDP/IP (see clause 6.2) requires a session control protocol to set-up and control of the individual media streams. For the transport of discrete media this specification adopts the use of HTTP/TCP/IP (see clause 6.3). In this case there is no need for a separate session set-up and control protocol since this is built into HTTP. This clause describes session set-up and control of continuous media.

5.3.2 RTSP

RTSP [5] shall be used for session set-up and session control. PSS clients and servers shall follow the rules for minimal on-demand playback RTSP implementations in appendix D of [5]. In addition to this:

- PSS servers and clients shall implement the DESCRIBE method (see clause 10.2 in [5]);

- PSS servers and clients shall implement the Range header field (see clause 12.29 in [5]).

5.3.3 SDP

5.3.3.1 General

RTSP requires a presentation description. SDP shall be used as the format of the presentation description for both PSS clients and servers. PSS servers shall provide and clients interpret the SDP syntax according to the SDP specification [6] and appendix C of [5]. The SDP delivered to the PSS client shall declare the media types to be used in the session using a codec specific MIME media type for each media. MIME media types to be used in the SDP file are described in clause 5.4 of the present document.

The SDP [6] specification requires certain fields to always be included in an SDP file. Apart from this a PSS server shall always include the following fields in the SDP:

- "a=control:" according to clauses C.1.1, C.2 and C.3 in [5];

- "a=range:" according to clause C.1.5 in [5];

- "a=rtpmap:" according to clause 6 in [6];

- "a=fmtp:" according to clause 6 in [6].

The bandwidth field in SDP can be used to indicate to the PSS client the amount of bandwidth that is required for the session and the individual media in the presentation. Therefore, a PSS server should include the "b=AS:" field in the SDP (both on the session and media level) and a PSS client shall be able to interpret this field. The bandwidth value shall indicate maximum net rates of media streams without lower level packetisation overhead

5.3.3.2 Additional SDP fields

The following Annex F-related media level SDP fields are defined for PSS:

- "a=x-predecbufsize:<size of the hypothetical pre-decoder buffer>"
  This gives the suggested size of the Annex F hypothetical pre-decoder buffer in bytes.

- a="x-initpredecbufperiod:<initial pre-decoder buffering period>"
  This gives the required initial pre-decoder buffering period specified according to Annex F. Values are interpreted as clock ticks of a 90-kHz clock. 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.

- "a=x-initpostdecbufperiod:<initial post-decoder buffering period>"
  This gives the required initial post-decoder buffering period specified according to Annex F. Values are interpreted as clock ticks of a 90-kHz clock.

If none of the attributes "a=x-predecbufsize:", "a=x-initpredecbufperiod:" and "a=x-initpostdecbufperiod:" is present, clients should not expect a packet stream according to Annex F. If at least one of the listed attributes is present, the transmitted video packet stream shall conform to Annex F. If any of the attributes is missing (assuming at least one is present) in an SDP description, clients should expect a default value for the attribute according to Annex F.

Editor's Note: The possibility to register our SDP attributes at IANA should be investigated.

5.4 MIME media types

For continuous media (speech, audio and video) the following MIME media types shall be used:

- AMR narrow band speech codec (see clause 7.2) MIME media type as defined in [11];

- AMR wide band speech codec (see clause 7.2) MIME media type as defined in [12];

- MPEG-4 AAC audio codec (see clause 7.3) MIME media type as defined in RFC 3016 [13].

- MPEG-4 video codec (see clause 7.4) MIME media type as defined in RFC 3016 [13];

- H.263 [22] video codec (see clause 7.4) MIME media type as defined in annex C, clause C.1 of the present document.

MIME media types for JPEG, GIF, PNG and XHTML can be used both in the "Content-type" field in HTTP and in the "type" attribute in SMIL 2.0. The following MIME media types shall be used for these media:

- JPEG (see clause 7.5) MIME media type as defined in [15];

- GIF (see clause 7.6) MIME media type as defined in [15];

- PNG (see sub clause 7.6) MIME media type as defined in [36];

- XHTML (see clause 7.8) MIME media type as defined in annex C clause C.2 of the present document.

MIME media type used for SMIL files shall be according to [31] and for SDP files according to [6].

Editor's Note: Need to add MIME types for vector graphics and synthetic audio.

6 Data transport

6.1 Packet based network interface

PSS clients and servers shall support an IP-based network interface for the transport of session control and media data. Control and media data are sent using TCP/IP [8] and UDP/IP [7]. An overview of the protocol stack can be found in figure 2 of the present document.

6.2 RTP over UDP/IP

The IETF RTP [9] and [10] provides a means for sending real-time or streaming data over UDP (see [7]). The encoded media is encapsulated in the RTP packets with media specific RTP payload formats. RTP payload formats are defined by IETF. RTP also provides a protocol called RTCP (see clause 6 in [9]) for feedback about the transmission quality.

RTP/UDP/IP transport of continuous media (speech ,audio and video) shall be supported.

For RTP/UDP/IP transport of continuous media the following RTP payload formats shall be used:

- AMR narrow band speech codec (see clause 7.2) RTP payload format according to [11];

- AMR wide band speech codec (see clause 7.2) RTP payload format according to [12];
- MPEG-4 AAC audio codec (see clause 7.3) RTP payload format according to RFC 3016 [13];
- MPEG-4 video codec (see clause 7.4) RTP payload format according to RFC 3016 [13];
- H.263 [22] video codec (see clause 7.4) RTP payload format according to RFC 2429 [14];

6.3 HTTP over TCP/IP

The IETF TCP provides reliable transport of data over IP networks, but with no delay guarantees. It is the preferred way for sending the scene description, text, bitmap graphics and still images. There is also need for an application protocol to control the transfer. The IETF HTTP [17] provides this functionality.

HTTP/TCP/IP transport shall be supported for:

- still images (see clause 7.5);
- bitmap graphics (see clause 7.6);
- text (see clause 7.8);
- scene description (see clause 8);
- presentation description (see clause 5.3.3).

6.4 Transport of RTSP

Transport of RTSP shall be supported according to RFC 2326 [5].

Editor's Note: Need to add transport for vector graphics and synthetic audio.

7 Codecs

7.1 General

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

7.2 Speech

The AMR codec shall be supported for narrow-band speech [18]. The AMR wideband speech codec [20] shall be supported when wideband speech working at 16 kHz sampling frequency is supported.

7.3 Audio

MPEG-4 AAC Low Complexity object type [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 may be supported.

7.3A Synthetic audio

The Scalable Polyphony MIDI (SP-MIDI) content format defined in Scalable Polyphony MIDI Specification [41] and the device requirements defined in Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP [42] should be supported.

Editor's Note: The working assumption for the new media type synthetic audio in Release 5 is as stated above. Further clarifications are needed for SP-MIDI MIME type, transport of synthetic audio (e.g. file format details), possible capability exchange attribute and the additional technical requirements introduced in contribution S4-010625 about the profile. The question whether SP-MIDI should be mandatory ("should" replaced by "shall") is under discussion in the PSM SWG. The proposals in contributions S4-010612 and S4-010625 have conflicting details but should both be considered as the starting point for further discussion. The references [41] and [42] can be found in S4-010612.

7.4 Video

ITU-T Recommendation H.263 [22] profile 0 level 10 shall be supported. This is the mandatory video codec for the PSS. In addition, PSS should support:

- H.263 [23] Profile 3 Level 10;
- MPEG-4 Visual Simple Profile Level 0, [24] and [25].

These two video codecs are optional to implement.

An optional video buffer model is given in Annex F of the present document.

NOTE: ITU-T Recommendation H.263 [22] baseline has been mandated to ensure that video-enabled PSS support a minimum baseline video capability and interoperability can be guaranteed (an H.263 [22] baseline bitstream can be decoded by both H.263 [22] and MPEG-4 decoders). It also provides a simple upgrade path for mandating more advanced codecs in the future (from both the ITU-T and ISO MPEG).

7.5 Still images

ISO/IEC JPEG [26] together with JFIF [27] 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 [26];
- progressive DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF2' [26].

7.6 Bitmap graphics

The following bitmap graphics codecs should be supported:

- GIF87a, [32];
- GIF89a, [33];
- PNG, [36].

7.7 Vector graphics

The 2D vector graphics format Scalable Vector Graphics (SVG) [40] should be supported.

Editor's Note: The question whether SVG should be mandatory ("should" replaced by "shall") is under discussion in the PSM SWG. The working assumption is SVG and the particular profile of SVG to support is also under discussion, e.g. SVG-Basic or SVG-Tiny.

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

7.8 Text

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

- text formatted according to XHTML Basic [28];
- 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 encoding shall be supported:

- UTF-8, [29];
- UCS-2, [30].

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.

8 Scene description

8.1 General

The 3GPP PSS use a subset of SMIL 2.0 [31] as format of the scene description. PSS clients and servers with support for scene descriptions shall support the 3GPP PSS4 SMIL Language Profile defined in clause 8.2. This profile is a subset of the SMIL 2.0 Language Profile, but a superset of the SMIL 2.0 Basic Language Profile. The present document also includes an informative Annex B that provides guidelines for SMIL content authors.

NOTE: The interpretation of this is not that all streaming sessions are required to use SMIL. For some types of sessions, e.g. consisting of one single continuous media or two media synchronised by using RTP timestamps, SMIL may not be needed.

8.2 3GPP PSS4 SMIL Language Profile

8.2.1 Introduction

3GPP PSS4 SMIL is a markup language based on SMIL Basic [31] and SMIL Scalability Framework.

3GPP PSS4 SMIL shall consist of the modules required by SMIL Basic Profile (and SMIL 2.0 Host Language Conformance) and additional MediaAccessibility, MediaDescription, MediaClipping, MetaInformation, PrefetchControl and EventTiming modules. All in all the following modules are included:

- SMIL 2.0 Content Control Modules – BasicContentControl, SkipContentControl and PrefetchControl
- SMIL 2.0 Layout Module -- BasicLayout
- SMIL 2.0 Linking Module -- BasicLinking
- SMIL 2.0 Media Object Modules – BasicMedia, MediaClipping, MediaAccessibility and MediaDescription
- SMIL 2.0 Metainformation Module -- Metainformation
- SMIL 2.0 Structure Module -- Structure
- SMIL 2.0 Timing and Synchronization Modules -- BasicInlineTiming, MinMaxTiming, BasicTimeContainers, RepeatTiming and EventTiming

8.2.2 Document Conformance

A conforming 3GPP PSS4 SMIL document shall be a conforming SMIL 2.0 document.

All 3GPP PSS4 SMIL documents use SMIL 2.0 namespace.

<smil xmlns="http://www.w3.org/2001/SMIL20/Language">

3GPP PSS4 SMIL documents may declare requirements using systemRequired attribute:

```
EXAMPLE 1:    <smil xmlns="http://www.w3.org/2001/SMIL20/Language"
                  xmlns:EventTiming="http://www.w3.org/2000/SMIL20/CR/EventTiming "
                  systemRequired="EventTiming">
```

Namespace URI http://www.3gpp.org/SMIL20/PSS4/ identifies the 3GPP PSS4 SMIL. Authors can use this URI to indicate requirement for exact 3GPP PSS4 SMIL semantics for a document or a subpart of a document:

```
EXAMPLE 2:    <smil xmlns="http://www.w3.org/2001/SMIL20/Language"
                  xmlns:pss4="http://www.3gpp.org/SMIL20/PSS4/"
                  systemReqzuired="pss4">
```

The content authors generally should choose not to include the PSS requirement in the document unless the SMIL document relies on PSS specific semantics that are not part of the W3C SMIL. The reason for this is that SMIL players that are not conforming 3GPP PSS user agents may not recognize the PSS4 URI and thus refuse to play the document.

8.2.3 User Agent Conformance

A conforming 3GPP PSS4 SMIL user agent shall be a conforming SMIL Basic User Agent.

A conforming user agent shall implement the semantics of the language as described in this document.

A conforming user agent shall recognize the URIs of all included SMIL 2.0 modules. It shall also recognize URI http://www.3gpp.org/SMIL20/PSS4/ as referring to all modules and semantics of 3GPP SMIL language.

8.2.4 3GPP SMIL Language Profile

3GPP PSS4 SMIL is based on SMIL 2.0 Basic language profile [31]. This chapter defines the content model and integration semantics of the included modules where they differ from those defined by SMIL Basic.

8.2.4.1 Content Control Modules

3GPP PSS4 SMIL shall include the content control functionality of the BasicContentControl, SkipContentControl and PrefetchControl modules of SMIL 2.0. PrefetchControl is not part of SMIL Basic and is an additional module in this profile.

All BasicContentControl attributes listed in the module specification shall be supported.

NOTE: The SMIL specification [31] defines that all functionality of PrefetchControl module is optional. This mean that even that PrefetchControl is mandatory user agents may implement semantics of PrefetchControl module only partially or not to implement them at all. PrefetchControl module adds the **prefetch** element to the content model of SMIL Basic **body, switch, par** and **seq** elements.

The **prefetch** element has the attributes defined by the PrefetchControl module (**mediaSize, mediaTime and bandwidth**), the **src** attribute, the BasicContentControl attributes and the **skip-content** attribute.

8.2.4.2 Layout Module

3GPP PSS4 SMIL shall use the BasicLayout module of SMIL 2.0 for spatial layout. The module is part of SMIL Basic.

Default values of the width and height attributes for root-layout shall be the dimensions of the device display area.

8.2.4.3 Linking Module

3GPP PSS4 SMIL shall use the SMIL 2.0 BasicLinking module for providing hyperlinks between documents and document fragments. This module is from SMIL Basic.

When linking to destinations outside the current document, implementations may ignore values "play" and "pause" of the ’sourcePlaystate’ attribute and values "new" and "pause" of the ’show’ attribute, instead using the semantics of values "stop" and "replace" respectively. When the values of ’sourcePlaystate’ and ’show’ are ignored the player may also ignore the ’sourceLevel’ attribute since it is of no use then

8.2.4.4 Media Object Modules

3GPP PSS4 SMIL shall include the media elements from the SMIL 2.0 BasicMedia module and attributes from the MediaAccessibility, MediaDescription and MediaClipping modules. MediaAccessibility, MediaDescription and MediaClipping modules are additions in this profile to the SMIL Basic.

See clause 5.4 for what are the mandatory and optional MIME types a 3GPP PSS4 SMIL player needs to support.

MediaClipping module adds to the profile the ability to address sub-clips of continuous media. MediaClipping module adds ’**clipBegin**’ and ’**clipEnd**´(and for compatibility '**clip-begin**' and '**clip-end**') attributes to all media elements.

MediaAccessibility module provides basic accessibility support for media elements. New attributes '**alt**', '**longdesc**' and '**readIndex**' are added to all media elements by this module. MediaDescription module is included by the MediaAccessibility module and adds '**abstract**', '**author**' and '**copyright**' attributes to media elements.

8.2.4.5 Metainformation Module

MetaInformation module of SMIL 2.0 shall be included to the profile. This module is addition in this profile to the SMIL Basic and provides a way to include descriptive information about the document content into the document.

This module adds **meta** and **metadata** elements to the content model of SMIL Basic **head** element.

8.2.4.6 Structure Module

The Structure module defines the top-level structure of the document. It's included by SMIL Basic.

8.2.4.7 Timing and Synchronization modules

The timing modules included in the 3GPP SMIL shall be BasicInlineTiming, MinMaxTiming, BasicTimeContainers, RepeatTiming and EventTiming. The EventTiming module is an addition in this profile to the SMIL Basic.

For 'begin' and 'end' attributes either single offset-value or single event-value shall be allowed. Offsets shall not be supported with event-values.

Event timing attributes that reference invalid IDs (for example elements that have been removed by the content control) shall be treated as being indefinite.

Supported event names and semantics shall be as defined by the SMIL 2.0 Language Profile. All user agents shall be able to raise the the following event types:

- activateEvent;
- beginEvent;
- endEvent.

Following SMIL 2.0 Language event types should be supported:

- focusInEvent;
- focusOutEvent;
- inBoundsEvent;
- outBoundsEvent;
- repeatEvent.

User agents shall ignore unknown event types and not treat them as errors.

Events do not bubble and shall be delivered to the associated media or timed elements only.

8.2.5 Content Model

This table shows the full content model and attributes of the 3GPP PSS4 SMIL profile. The attribute collections used are defined by SMIL Basic ([31], SMIL Host Language Conformance requirements, chapter 2.4). Changes to the SMIL Basic are shown in **bold**.

Element		
	Elements	**Attributes**
smil	head, body	COMMON-ATTRS, CONTCTRL-ATTRS, xmlns
head	layout, switch, **meta**, **metadata**	COMMON-ATTRS
body	TIMING-ELMS, MEDIA-ELMS, switch, a, **prefetch**	COMMON-ATTRS
layout	root-layout, region	COMMON-ATTRS, CONTCTRL-ATTRS, type
root-layout	EMPTY	COMMON-ATTRS, backgroundColor, height, width, skip-content
region	EMPTY	COMMON-ATTRS, backgroundColor, bottom, fit, height, left, right, showBackground, top, width, z-index, skip-content, regionName
ref, animation, audio, img, video, text, textstream	area	COMMON-ATTRS, CONTCTRL-ATTRS, TIMING-ATTRS, repeat, region, MEDIA-ATTRS, **clipBegin(clip-begin), clipEnd(clip-end), alt, longDesc, readIndex, abstract**, **author**, **copyright**
a	MEDIA-ELMS	COMMON-ATTRS, LINKING-ATTRS
area	EMPTY	COMMON-ATTRS, LINKING-ATTRS, TIMING-ATTRS, repeat, shape, coords, nohref
par, seq	TIMING-ELMS, MEDIA-ELMS, switch, a, **prefetch**	COMMON-ATTRS, CONTCTRL-ATTRS, TIMING-ATTRS, repeat
switch	TIMING-ELMS, MEDIA-ELMS, layout, a, **prefetch**	COMMON-ATTRS, CONTCTRL-ATTRS
prefetch	**EMPTY**	**COMMON-ATTRS, CONTCTRL-ATTRS, mediaSize, mediaTime, bandwidth, src, skip-content**
meta	**EMPTY**	**COMMON-ATTRS, content, name, skip-content**
metadata	**EMPTY**	**COMMON-ATTRS, skip-content**

9 Interchange format for MMS

9.1 General

The MPEG-4 file format [34] is mandated in [35] 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.

Additionally, the MPEG-4 file format can be used for the storage in the servers and the "hint track" mechanism can be used for the preparation for streaming.

The clause 9.2 of the present document gives the necessary requirements to follow for the MPEG-4 file format used in MMS. These requirements will guarantee PSS to interwork with MMS as well as the MPEG-4 file format to be used internally within the MMS system. For PSS servers not interworking with MMS there is no requirement to follow these guidelines.

9.2 MPEG-4 file format guidelines

9.2.1 Registration of non-ISO codecs

How to include the non-ISO code streams AMR narrow-band speech and H.263 encoded video in MP4 files is described in annex D of the present document.

9.2.2 Hint tracks

The hint tracks are a mechanism that the server implementation may choose to use in preparation for the streaming of media content contained in MP4 files. However, it should be observed that the usage of the hint tracks is an internal implementation matter for the server, and it falls outside the scope of the present document.

9.2.3 Self-contained MP4 files

All media in the MP4 file shall be self-contained, i.e. there shall not be referencing to external media data from inside the MP4 file.

9.2.4 MPEG-4 systems specific elements

Tracks relative to MPEG-4 system architectural elements (e.g. BIFS scene description tracks or OD Object descriptors) are optional and shall be ignored. The adoption of the MPEG-4 file format does not imply the usage of MPEG-4 systems architecture. The receiving terminal is not required to implement any of the specific MPEG-4 system architectural elements.

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.

Table A.1: Overview of fields in SDP

Type	Description		Requirement according to [6]	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
Z	Time zone adjustments		O	O
K	Encryption key		O	O
A	Session attributes	control	O	R
		range	O	R
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.				

Editor's Note: Table A.1 should be updated with the optional Annex F related attributes after SA has approved the pending CR on Annex A.

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
a=range:npt=0-45.678
b=AS:128
t=0 0
m=video 1024 RTP/AVP 96
a=rtpmap:96 H263-2000/90000
a=fmtp:96 profile=3;level=10
a=control:rtsp;//mediaserver.com/movie
a=recvonly
b=AS:128
```

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

_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.3 Capability exchange

A.3.1 Overview

Clause A.3 provides detailed information about the structure and exchange of device capability descriptions for the PSS. It complements the normative part contained in clause 5.2 of the present document.

The functionality is sometimes referred to as capability exchange. Capability exchange in PSS uses the CC/PP [37] framework and reuse parts of the CC/PP application UAProf [38].

To facilitate server-side content negotiation for streaming, the PSS server needs to have access to a description of the specific capabilities of the mobile terminal i.e. the device capability description. The device capability description contains a number of attributes. During the set-up of a streaming session the PSS server can use the description to provide the mobile terminal with the correct type of multimedia content. Concretely, it is envisaged that servers use information about the capabilities of the mobile terminal to decide which stream(s) to provision to the connecting terminal. For instance, the server could compare the requirements on the mobile terminal for multiple available variants of a stream with the actual capabilities of the connecting terminal to determine the best-suited stream(s) for that particular terminal. A similar mechanism could also be used for other types of content.

A device capability description contains a number of device capability attributes. In the present document they are referred to as just attributes. The current version of PSS does not include a definition of any specific user preference attributes. Therefore we use the term device capability description. However, it should be noted that even though no specific user preference attributes are included, simple tailoring to the preferences of the user could be achieved by temporarily overrides of the available attributes. E.g. if the user for a particular session only would like to receive mono sound even though the terminal is capable of stereo, this can be accomplished by providing an override for the "AudioChannels" attribute. It should also be noted that the extension mechanism defined would enable an easy introduction of specific user preference attributes in the device capability description if needed.

The term device capability profile or profile is sometimes used instead of device capability description to describe a description of device capabilities and/or user preferences. The three terms are used interchangeably in the present document.

Figure A.1 illustrates how capability exchange in PSS is performed. In the simplest case the mobile terminal informs the PSS server(s) about its identity so that the latter can retrieve the correct device capability profile(s) from the device profile server(s). For this purpose, the mobile terminal adds one or several URLs to RTSP and/or HTTP protocol data units that it sends to the PSS server(s). These URLs point to locations on one or several device profile servers from where the PSS server should retrieve the device capability profiles. This list of URLs is encapsulated in RTSP and HTTP protocol data units using additional header field(s). The list of URLs is denoted URLdesc. The mobile terminal may additional complement the URLdesc with extra attributes or overrides for attributes already defined in the profile(s) located at URLdesc. This information is denoted Profdiff. As URLdesc, Profdiff is encapsulated in RTSP and HTTP protocol data units using additional header field(s).

The device profile server in Figure A.1 is the logical entity that stores the device capability profiles. The profile needed for a certain request from a mobile terminal may be stored on one or several such servers. A terminal manufacturer or a software vendor could maintain a device profile server to provide device capability profiles for its products. It would also be possible for an operator to manage a device profile server for its subscribers and then e.g. enable the subscriber to make user specific updates to the profiles. The device profile server provides device capability profiles to the PSS server on request.

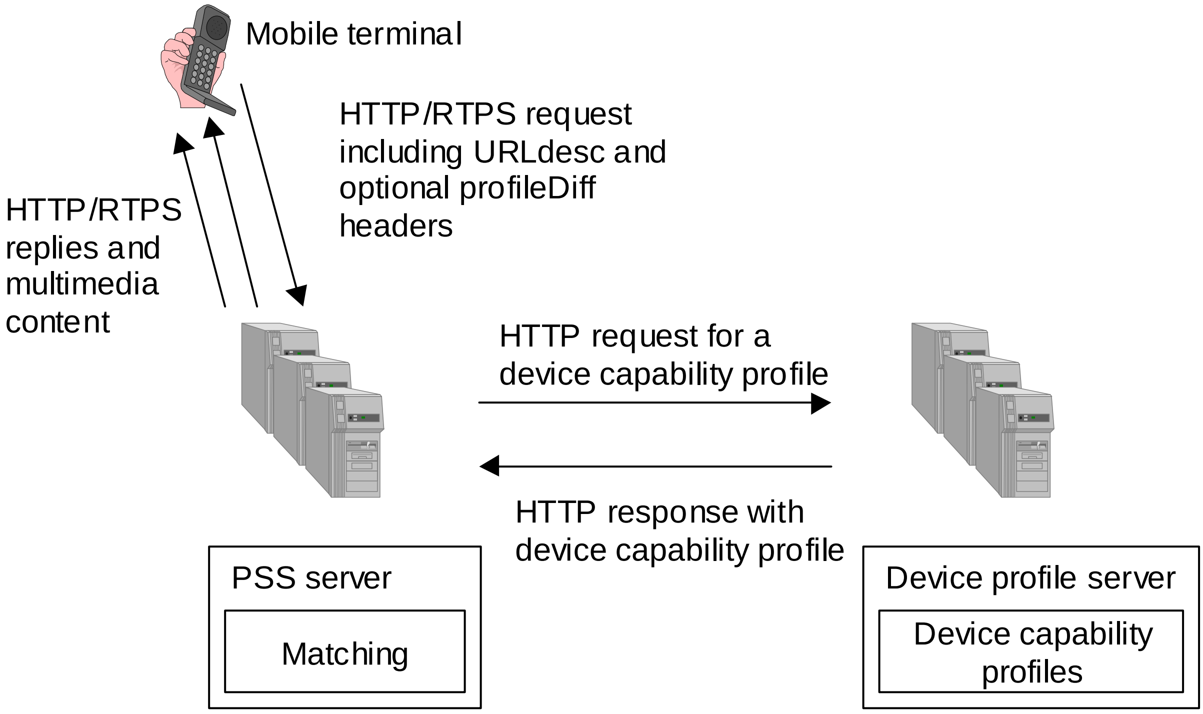

Mobile terminal
HTTP/RTPS request including URLdesc and optional profileDiff headers
HTTP/RTPS replies and multimedia content
HTTP request for a device capability profile
HTTP response with device capability profile
PSS server
Matching
Device profile server
Device capability profiles

Figure A.1: Functional components in PSS capability exchange

The PSS server is the logical entity that provides multimedia streams and other, static content (e.g. SMIL documents, images, and graphics) to the mobile terminal (see Figure A.1). A PSS application might involve multiple PSS servers, e.g. separate servers for multimedia streams and for static content. A PSS server handles the matching process. Matching is a process that takes place in the PSS servers (see Figure A.1). The device capability profile is compared with the content descriptions at the server and the best fit is delivered to the client

A.3.2 Scope of the specification

The following bullet list describes what is considered to be within the scope of the specification for capability exchange in PSS.

- Definition of the structure for the device capability profiles, see clause A.3.3.
- Definition CC/PP vocabularies, see clause A.3.4.
  - Reference to a set of device capability attributes for multimedia content retrieval applications that have already been defined by UAProf [38]. The purpose of this reference is to point out which attributes are useful for the PSS application.
  - Definition of a set of device capability attributes specifically for PSS applications that are missing in UAProf.

- It is important to define an extension mechanism to easily add attributes since it is not possible to cover all attributes from the beginning. The extension mechanism is described in clause A.3.5.
- The structure of URLdesc, Profdiff and their interchange is described in clause A.3.6.
- Protocols for the interchange of device capability profiles between the PSS server and the device profile server is defined in clause 5.2.7.

The specification does not include:

- rules for the matching process on the PSS server. These mechanisms should be left to the implementations. For interoperability, only the format of the device capability description and its interchange is relevant. Not the mechanism how to compare the device capability profile of the terminal with the descriptions of the available content and perform the selection.

- definition of specific user preference attributes. It is very difficult standardise such attributes since it is dependent on the type of personalised services one would like to offer the user. The extensible descriptions format and exchange mechanism proposed in this document provide the means to create and exchange such attributes if needed in the future. However, as explained in clause A.3.1 limited tailoring to the preferences of the user could be achieved by temporarily overrides of the available attributes in the vocabularies already defined for PSS. The available attributes are described in clause 5.2.3 of the present document.

- requirements for caching of device capability profiles on the PSS server. In UAProf, a content server can cache the current device capability profile for a given WSP session. This feature relies on the presence of WSP sessions. Caching significantly increases the complexity of both the implementations of the mobile terminal and the server. However, HTTP is used between the PSS server and the device profile server. For this exchange, normal content caching provisions as defined by HTTP apply and the PSS server may utilise this to speed up the session set-up (see clause 5.2.7)

- intermediate proxies. This feature is considered not relevant in the context of PSS applications.

A.3.3 The device capability profile structure

A device capability profile is a description of the capabilities of the device and possibly also the preferences of the user of that device. It can be used to guide the adaptation of content presented to the device. A device capability profile for PSS is a RDF [39] document that follows the structure of the CC/PP framework [37] and the CC/PP application UAProf [38]. The terminology of CC/PP is used in this text and therefore briefly described here.

Attributes are used for specifying the device capabilities and user preferences. A set of attribute names, permissible values and semantics constitute a CC/PP vocabulary. A RDF schema defines a vocabulary. The syntax of the attributes is defined in the schema but also, to some extent, the semantics. A profile is an instance of a schema and contains one or more attributes from the vocabulary. Attributes in a schema are divided into components distinguished by attribute characteristics. In the CC/PP specification it is anticipated that different applications will use different vocabularies. According to the CC/PP framework a hypothetical profile might look like Figure A.2. A further illustration of how a profile might look like is given in the example in clause A.3.7.

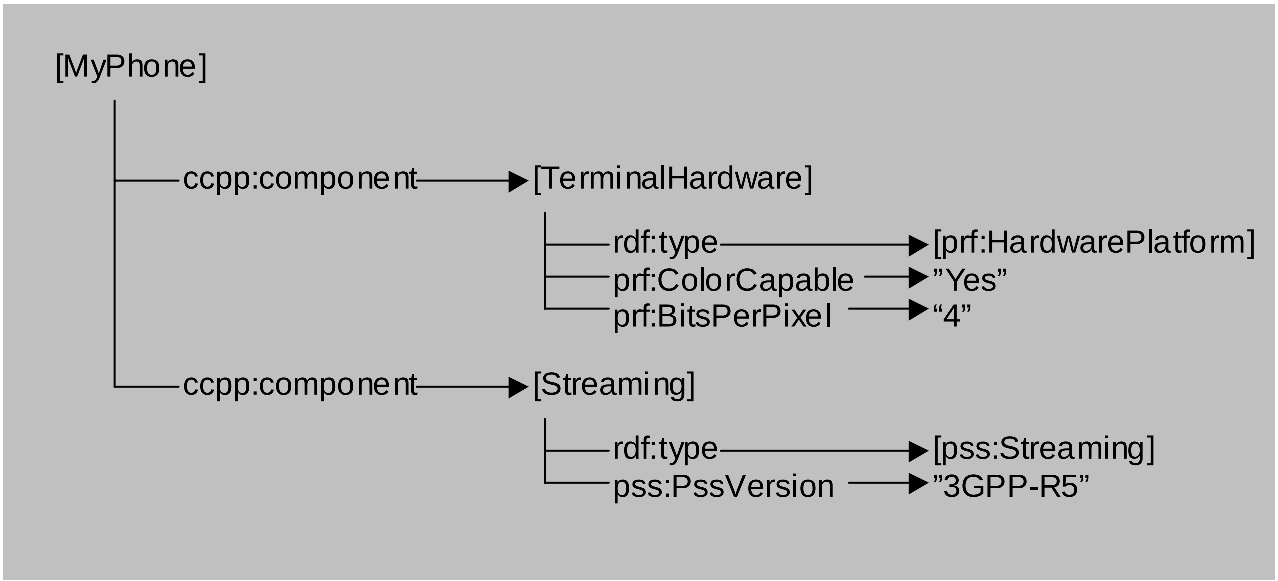


Figure A.2: Illustration of the profile structure

A CC/PP schema is extended through the introduction of new attribute vocabularies and a device capability profile can use attributes drawn from an arbitrary number of different vocabularies. Each vocabulary is associated with a unique XML namespace. This mechanism makes it possible to reuse attributes from other vocabularies.

Attributes of a component can be included directly or may be specified by a reference to a CC/PP default profile. Resolving a profile that includes a reference to a default profile is time-consuming. When the PSS server receives the profile from a device profile server the final attribute values can not be determined until the default profile has been requested and received. Support for defaults is required by the CC/PP specification [37]. Due to these problems, there is a recommendation made in clause 5.2.6 to not use the CC/PP defaults element in PSS device capability profile documents.

A.3.4 CC/PP Vocabularies

A CC/PP vocabulary shall according to CC/PP and UAProf include

- A RDF schema for the vocabulary based on the CC/PP schema.
- A description of the semantics/type/resolution rules/sample values for each attribute.
- A unique namespace shall be assigned to each version of the profile schema.

Additional information that could be included in the profile schema

- A description about the profile schema, i.e. the purpose of the profile, how to use it, when to use it etc.
- A description of extensibility. How to handle future extensions of the profile schema.

A device capability profile can use an arbitrary number of vocabularies and thus it is possible to reuse attributes from other vocabularies by simply reference the corresponding namespaces. The focus of the PSS vocabulary is content formatting which overlaps the focus of the UAProf vocabulary. UAProf is specified by WAP Forum and is an architecture and vocabulary/schema for capability exchange in the WAP environment. Since there are attributes in the UAProf vocabulary suitable for streaming applications these are reused and combined with a PSS application specific streaming component. This makes the PSS vocabulary an extension vocabulary to UAProf. The CC/PP specification encourages reuse of attributes from other vocabularies. To avoid confusion, the same attribute name should not be used in different vocabularies. In clause 5.2.3.3 a number of attributes from UAProf [38] are recommended for PSS. The PSS base vocabulary is defined in clause 5.2.3.2.

A profile is allowed to instantiate a subset of the attributes in the vocabularies and no specific attributes are required but insufficient description may lead to content unable to be shown by the client.

A.3.5 Principles of extending a schema/vocabulary

The use of RDF enables an extensibility mechanism for CC/PP-based schemas that addresses the evolution of new types of devices and applications. The PSS profile schema specification is going to provide a base vocabulary but in the future new usage scenarios might have need for expressing new attributes. This is the reason why there is a need to specify how extensions of the schema will be handled. If the TSG responsible for the present document updates the base vocabulary schema a new unique namespace will be assigned to the updated schema. In another scenario the TSG may decide to add a new component containing specific user related attributes. This new component will be assigned a new namespace and it will not influence the base vocabulary in any way. If other organisations or companies make extensions this can be either as a new component or as attributes added to the existing base vocabulary component where the new attributes uses a new namespace. This ensures that third parties can define and maintain their own vocabularies independently from the PSS base vocabulary.

A.3.6 Signalling of profile information between client and server

The concept of URLdesc and Profdiff was introduced in clause A.3.1. The URLdesc is a list of URLs that point to locations on device profile servers from where the PSS server retrieves suitable device capability profiles. The Profdiff contains additional capability description information; e.g. overrides for certain attribute values. Both URLdesc and Profdiff are encapsulated in RTSP and HTTP messages using additional header fields. This can be seen in Figure A.1.

In clause 9.1 of [38] three new HTTP headers are defined that can be used to implement the desired functionality: "x-wap-profile", "x-wap-profile-diff" and "x-wap-profile-warning". These headers are reused in PSS for both HTTP and RTSP.

- The "x-wap-profile" is a request header that contains a list of absolute URLs to device capability descriptions and profile diff names. The profile diff names correspond to additional profile information in the "x-wap-profile-diff" header.
- The "x-wap-profile-diff" is a request header that contains a subset of a device capability profile.
- The "x-wap-profile-warning" is a response header that contains error codes explaining to what extent the server has been able to match the terminal request.

Clause 5.2.5 of the present document defines this exchange mechanism.

It is left to the mobile terminal to decide when to send x-wap-profile headers. The mobile terminal could send the "x-wap-profile" and "x-wap-profile-diff" headers with each RTSP DESCRIBE and/or with each RTSP SETUP request. Sending them in the RTSP DESCRIBE request is useful for the PSS server to be able to make a better decision which presentation description to provision to the client. Sending the "x-wap-profile" and "x-wap-profile-diff" headers with an HTTP request is useful whenever the mobile terminal requests some multimedia content that will be used in the PSS application. For example it can be sent with the request for a SMIL file and the PSS server can see to that the mobile terminal receives a SMIL file which is optimised for the particular terminal. Clause 5.2.5 of the present document gives recommendations for when profile information should be sent.

It is up to the PSS server to retrieve the device capability profiles using the URLs in the "x-wap-profile" header. The PSS server is also responsible to merge the profiles then received. If the "x-wap-profile-diff" header is present it must also merge that information with the retrieved profiles. This functionality is defined in clause 5.2.6.

It should be noted that it is up the implementation of the mobile terminal what URLs to send in the "x-wap-profile" header. For instance, a terminal could just send one URL that points to a complete description of its capabilities. Another terminal might provide one URL that points to description of the terminal hardware. A second URL that points to description of a particular software version of the streaming application, and a third URL that points to the description of a hardware or software plug-in that is currently added to the standard configuration of that terminal. From this example it becomes clear that sending URLs from the mobile terminal to the server is good enough not only for static profiles but that it can also handle re-configurations of the mobile terminal such as software version changes, software plug-ins, hardware upgrades, etc.

As described above the list of URLs in the x-wap-profile header is a powerful tool to handle dynamic changes of the mobile terminal. The "x-wap-profile-diff" header could also be used to facilitate the same functionality. To use the "x-wap-profile-diff" header to e.g. send a complete profile (no URL present at all in the "x-wap-profile header") or updates as a result of e.g. a hardware plug-in is not recommended unless some compression scheme is applied over the air-interface. The reason is of course that the size of a profile may be large.

A.3.7 Example of a PSS device capability description

```
<?xml version=”1.0”?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#"
         xmlns:prf="http://www.wapforum.org/profiles/UAPROF/ccppschema-20010430#"
         xmlns:pss5=”http://www.3gpp.org/profiles/PSS/ccppschema-200108#”>

  <rdf:Description rdf:about=”http://www.ericsson.com/Phones/Phone007”>

    <ccpp:component>
      <rdf:Description ID=”HardwarePlatform”>
      <rdf:type rdf:resource=”http://www.wapforum.org/profiles/UAPROF/ccppschema-
                              20010430#HardwarePlatform”>
        <prf:BitsPerPixel>4</prf:BitsPerPixel>
        <prf:ColorCapable>Yes</prf:ColorCapable>
        <prf:PixelAspectRatio>1x2</prf:PixelAspectRatio>
        <prf:PointingResolution>Pixel</prf:PointingResolution>
        <prf:ScreenSize>73x50 </prf:ScreenSize>
        <prf:SoundOutputCapable>Yes</prf:SoundOutputCapable>
        <prf:Model>Phone007</prf:Model>
        <prf:Vendor>Ericsson</prf:Vendor>
      </rdf:Description>
    </ccpp:component>
    <ccpp:component>
      <rdf:Description ID=”SoftwarePlatform”>
      <rdf:type rdf:resource=”http://www.wapforum.org/profiles/UAPROF/ccppschema-
                              20010430#SoftwarePlatform”>
        <prf:CcppAccept-Charset>
          <rdf:Bag>
            <rdf:li>UTF-8</rdf:li>
            <rdf:li>ISO-10646-UCS-2</rdf:li>
          </rdf:Bag>
        </prf:CcppAccept-Charset>
        <prf:CcppAccept-Encoding>
          <rdf:Bag>
            <rdf:li>base64</rdf:li>
            <rdf:li>quoted-printable</rdf:li>
          </rdf:Bag>
        </prf:CcppAccept-Encoding>
        <prf:CcppAccept-Language>
          <rdf:Seq>
            <rdf:li>en</rdf:li>
          </rdf:Seq>
        </prf:CcppAccept-Language>
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description ID=”Streaming”>
      <rdf:type rdf:resource=” http://www.3gpp.org/profiles/PSS/ccppschema-200108#Streaming”>
        <pss5:AudioChannels>Stereo</pss5:AudioChannels>
        <pss5:VideoPreDecoderBufferSize>30720</pss5:VideoPreDecoderBufferSize>
        <pss5:VideoInitialPostDecoderBufferingPeriod>0</pss5:VideoInitialPostDecoderBufferingPeriod>
        <pss5:PssAccept>
          <rdf:Bag>
            <rdf:li>audio/AMR-WB</rdf:li>
            <rdf:li>video/MP4V-ES</rdf:li>
          </rdf:Bag>
        </pss5:PssAccept>
        <pss5:PssAccept-Subset>
          <rdf:Bag>
            <rdf:li>JPEG Baseline</rdf:li>
          </rdf:Bag>
        </pss5:PssAccept-Subset>
        <pss5:PssVersion>3GPP-R5</pss5:PssVersion>
        <pss5:RenderingScreenSize>70x40</pss5:RenderingScreenSize>
        <pss5:SmilBaseSet>SMIL-3GPP-R4</pss5:SmilBaseSet>
        <pss5:SmilModules>
          <rdf:Bag>
            <rdf:li>BasicTransitions</rdf:li>
            <rdf:li>MulitArcTiming</rdf:li>
          </rdf:Bag>
        </pss5:SmilModules>
      </rdf:Description>
    </ccpp:component>

  </rdf:Description>
</rdf:RDF>
```

Annex B (informative): SMIL authoring guidelines

B.1 General

This is an informative annex for SMIL presentation authors. Authors can expect that PSS clients can handle the SMIL module collection defined in clause 8.2, with the restrictions defined in this Annex. When creating SMIL documents the author is recommended to consider that terminals may have small displays and simple input devices. The media types and their encoding included in the presentation should be restricted to what is described in clause 7 of the present document. Considering that many mobile devices may have limited software and hardware capabilities, the number of media to be played simultaneous should be limited. For example, many devices will not be able to handle more than one video sequence at the time.

B.2 BasicLinking

The Linking Modules define elements and attributes for navigational hyperlinking, either through user interaction or through temporal events. The BasicLinking module defines the a and area elements for basic linking:

a Similar to the "a" element in HTML it provides a link from a media object through the href attribute (which contains the URI of the link's destination). The "a" element includes a number of attributes for defining the behaviour of the presentation when the link is followed.

area Whereas the a element only allows a link to be associated with a complete media object, the area element allows links to be associated with spatial and/or temporal portions of a media object.

The area element may be useful for enabling services that rely on interactivity where the display size is not big enough to allow the display of links alongside a media (e.g. QCIF video) window. Instead, the user could, for example, click on a watermark logo displayed in the video window to visit the company website.

Even if the area element may be useful some mobile terminals will not be able to handle area elements that include multiple selectable regions within an area element. One reason for this could be that the terminals do not have the appropriate user interface. Such area elements should therefore be avoided. Instead it is recommended that the "a" element be used. If the "area" element is used, the SMIL presentation should also include alternative links to navigate through the presentation; i.e. the author should not create presentations that rely on that the player can handle "area elements.

B.3 BasicLayout

The "fit" attribute defines how different media should be fitted into their respective display regions.

The rendering and layout of some objects on a small display might be difficult and all mobile devices may not support features such as scroll bars; in addition, the root-layout window may represent the full screen of the display. Therefore "fit=scroll" should not be used.

Due to hardware restrictions in mobile devices, operations such that scaling of a video sequence, or even images, may be very difficult to achieve. According to the SMIL 2.0 specification SMIL players may in these situations clip the content instead. To be sure of that the presentation is displayed as the author intended, content should be encoded in a size suitable for the terminals intended and it is recommended to use "fit=hidden".

B.4 EventTiming

The two attributes "endEvent" and "repeatEvent" in the EventTiming module may cause problems for a mobile SMIL player. The end of a media element triggers the "endEvent". In the same way the "repeatEvent" occurs when the second and subsequent iterations of a repeated element begin playback. Both these events rely on that the SMIL player receives information about that the media element has ended. One example could be when the end of a video sequence initiates the event. If the player has not received explicit information about the duration of the video sequence, e.g. by the "dur" attribute in SMIL or by some external source as the "a=range" field in SDP. The player will have to rely on the RTCP BYE message to decide when the video sequence ends. If the RTCP BYE message is lost, the player will have problems initiate the event. For these reasons is recommended that the "endEvent" and "repeatEvent" attributes are used with care, and if used the player should be provided with some additional information about the duration of the media element that triggers the event. This additional information could e.g. be the "dur" attribute in SMIL or the "a=range" field in SDP.

The "inBoundsEvent" and "outOfBoundsEvent" attributes assume that the terminal has a pointer device for moving the focus to within a window (i.e. clicking within a window). Not all terminals will support this functionality since they do not have the appropriate user interface. Hence care should be taken in using these particular event triggers.

B.5 MetaInformation

Authors are encouraged to make use of meta data whenever providing such information to the mobile terminal appears to be useful. However, they should keep in mind that some mobile terminals will parse but not process the meta data.

Furthermore, authors should keep in mind that excessive use of meta data will substantially increase the file size of the SMIL presentation that needs to be transferred to the mobile terminal. This may result in longer set-up times.

B.6 XML entities

Entities are a mechanism to insert XML fragments inside an XML document. Entities can be internal, essentially a macro expansion, or external. Use of XML entities in SMIL presentations is not recommended, as many current XML parsers do not fully support them.

B.7 XHTML Basic

When rendering texts in a SMIL presentation, authors are able to use XHTML Basic that contains eleven modules. However, some of the modules include non-text information. When referring to an XHTML Basic document from a SMIL document, authors should use only *the required XHTML Host Language modules* : Structure Module, Text Module, Hypertext Module and List Module. The use of the Image Module, in particular, should not be used. Images and other non-text contents should be included in the SMIL document.

Note: An XHTML file Including a module which is not part of the XHTML Host Language modules may not be shown as intended.

Annex C (normative):
MIME media types

C.1 MIME media type H263-2000

MIME media type name: video
MIME subtype name: H263-2000

Required parameters: None

Optional parameters:
profile: H.263 profile number, in the range 0 through 8, specifying the supported H.263 annexes/subparts.
level: Level of bitstream operation, in the range 0 through 99, specifying the level of computational complexity of the decoding process. When no profile and level parameters are specified, Baseline Profile (Profile 0) level 10 are the default values.

The profile and level specifications can be found in [23]. Note that the RTP payload format for H263-2000 is the same as for H263-1998 and is defined in [14], but additional annexes/subparts are specified along with the profiles and levels.

NOTE: The above text will be replaced with a reference to the RFC describing the H263-2000 MIME media type as soon as this becomes available.

C.2 MIME media type xhtml+xml

MIME media type name: application
MIME subtype name: xhtml+xml

Required parameters: none

Optional parameters:
charset: This parameter has identical semantics to the charset parameter of the "application/xml" media type as specified in [16].

NOTE: The above text will be replaced with a reference to the RFC describing the xhtml+xml MIME media type as soon as this becomes available.

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 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 and H.263 are given in clauses D.5 to D.8.

AMR data is stored in the stream according to clause 8 of [11].

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.

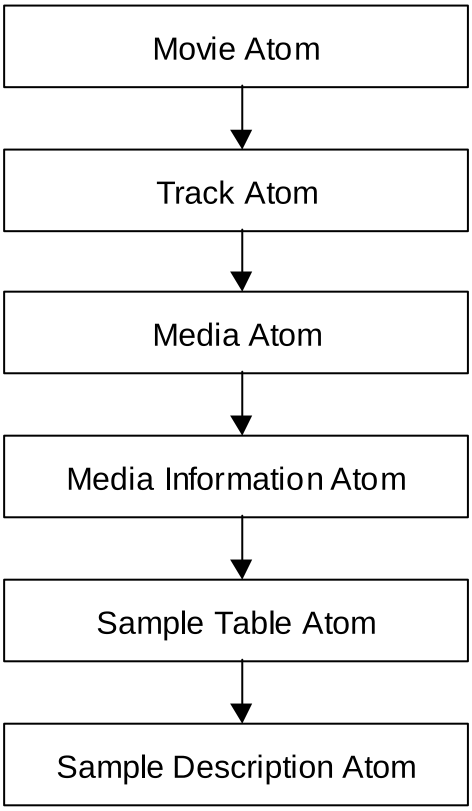


Figure D.1: MP4 Atom Structure Hierarchy

The Sample Description Atom can have one or more SampleDescriptionEntry fields. Valid Sample Description Entry atoms already defined for MP4 are AudioSampleEntry, VideoSampleEntry, HintSampleEntry and MPEGSampleEntry Atoms. The Sample DescriptionEntry Atoms for AMR and 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 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

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)		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)		0
Reserved_4	Const unsigned int(32)		0x014000f0
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)		0
Reserved_2	Const unsigned int(16)		24
Reserved_2	Const int(16)		-1
ESDAtom		Elementary stream descriptor for this stream.	

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

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)		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)		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 stream descriptor for this stream.	

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

D.5 AMRSampleEntry atom

The atom type of the AMRSampleEntry Atom shall be 'samr'.

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

Table D.4: AMRSampleEntry fields

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		’samr’
Reserved_6	Unsigned int(8)		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)		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		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. The 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 AMRSampleEntry Atom is defined as follows:

H263SampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_16

Reserved_4

Reserved_4

Reserved_4

Reserved_4

Reserved_2

Reserved_32

Reserved_2

Reserved_2

DecoderSpecificInfo

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)		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)		0
Reserved_4	Const unsigned int(32)		0x014000f0
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)		0
Reserved_2	Const unsigned int(16)		24
Reserved_2	Const int(16)		-1
DecoderSpecificInfo		Information specific to the 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 DecoderSpecificInfo field structure for H.263 is described in clause D.8.

D.7 DecoderSpecificInfo field for AMRSampleEntry atom

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

Table D.6: The DecoderSpecificInfo fields for AMRSampleEntry

Field	Type	Details	Value
DecSpecificInfoTag	Bit(8)		0x05
SizeOfDecSpecificInfo	Unsigned int(32)		
DecSpecificInfo	AMRDecSpecStruc	Structure which holds the AMR 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.

DecSpecificInfo: the structure where the AMR 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’.

decoder_version: version of the decoder which created the AMR stream being stored, the value is set to 0 if version has no importance.

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 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. 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 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 field for H263SampleEntry atom

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

The DecoderSpecificInfo for H263 is composed of the following fields.

Table D.7: The DecoderSpecificInfo fields H263SampleEntry

Field	Type	Details	Value
DecSpecificInfoTag	Bit(8)		0x05
SizeOfDecSpecificInfo	Unsigned int(32)		
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.

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'.

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.

Annex E (normative): RDF schema for the PSS base vocabulary

For the purpose of determining whether a profile conforms to this specification, the entire schema, including comments, is considered normative.

Editor's Note: Schema is to be defined and verified later. Must be available at an open and persistent URI, e.g. http://www.3gpp.org/profiles/PSS/ccppschema.

Annex F (normative): Buffering of video

F.1 Introduction

This annex describes video-buffering requirements in the PSS. As defined in clause 7.4 of the present document, support for the annex is optional and may be signalled in the PSS capability exchange and in the SDP. This is described in clause 5.2 and clause 5.3.3 of the present document. When the annex is in use, the content of the annex is normative. In other words, PSS clients shall be capable of receiving an RTP packet stream that complies with the specified buffering model and PSS servers shall verify that the transmitted RTP packet stream complies with the specified buffering model.

F.2 PSS server buffering verifier

The PSS server buffering verifier is specified according to the PSS buffering model. The model is based on two buffers and two timers. The buffers are called the hypothetical pre-decoder buffer and the hypothetical post-decoder buffer. The timers are named the decoding timer and the playback timer. The model is presented below.

1. The buffers are initially empty.

2. A PSS Server adds each transmitted RTP packet having video payload to the pre-decoder buffer immediately when it is transmitted. All protocol headers at RTP or any lower layer are removed.

3. Data is not removed from the pre-decoder buffer during a period called the initial pre-decoder buffering period. The period starts when the first RTP packet is added to the buffer.

4. When the initial pre-decoder buffering period has expired, the decoding timer is started from a position indicated in the previous RTSP PLAY request.

5. Removal of a video frame is started when both of the following two conditions are met: First, the decoding timer has reached the scheduled playback time of the frame. Second, a minimum interval since the beginning of the removal of the previous video frame has passed. The minimum interval is decided in the session set-up. Data is removed at a maximum allowable rate decided in the session set-up. Most often, the minimum frame interval and the maximum allowable removal rate are defined by means of video coding profiles and levels. For example, H.263 Level 10 requires support for minimum frame interval of 1001/15000 seconds and bit-rates up to 64000 bits per second. When the coded video frame has been removed from the pre-decoder buffer entirely, the corresponding uncompressed video frame is located into the post-decoder buffer.

6. Data is not removed from the post-decoder buffer during a period called the initial post-decoder buffering period. The period starts when the first frame has been placed into the post-decoder buffer.

7. When the initial post-decoder buffering period has expired, the playback timer is started from the position indicated in the previous RTSP PLAY request.

8. A frame is removed from the post-decoder buffer immediately when the playback timer reaches the scheduled playback time of the frame.

9. Each RTSP PLAY request resets the PSS buffering model to its initial state.

A PSS server shall verify that a transmitted RTP packet stream complies with the following requirements:

- The PSS buffering model shall be used with the default or signalled initial pre-decoder buffering period and with the default or signalled initial post-decoder buffering period.

- The occupancy of the hypothetical pre-decoder buffer shall not exceed the default or signalled buffer size.

- Each frame shall be inserted into the hypothetical post-decoder buffer before or on its scheduled playback time.

- The default initial pre-decoder buffering period is 1 second.

- The default initial post-decoder buffering period is zero.

- The default size of the hypothetical pre-decoder buffer is defined according to the maximum video bit-rate according to the table below:

Maximum video bit-rate	**Default size of the hypothetical pre-decoder buffer**
65536 bits per second	20480 bytes
131072 bits per second	40960 bytes
Undefined	51200 bytes

  The maximum video bit-rate can be signalled in the media-level bandwidth attribute of SDP as defined in clause 5.3.3 of this document. If the video-level bandwidth attribute was not present in the presentation description, the maximum video bit-rate is defined according to the video coding profile and level in use.

- The size of the hypothetical post-decoder buffer is an implementation-specific issue. The buffer size can be estimated from the maximum output data rate of the decoders in use and from the initial post-decoder buffering period.

F.3 PSS client buffering requirements

The PSS client shall be capable of receiving an RTP packet stream that complies with the PSS server buffering verifier, when the RTP packet stream is carried over a constant-delay reliable transmission channel. Furthermore, the video decoder of the PSS client, which may include handling of post-decoder buffering, shall output frames at the correct rate defined by the RTP time-stamps of the received packet stream.

Editor's Note: The text above needs to be updated to reflect that it is optional for the client to support the functionality in question. A similar wording as in clause F.1 could be used.

Annex G~~E~~ (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
03-2001	11	SP-010094			Version for Release 4		4.0.0
09-2001	13	SP-010457	001	1	3GPP PSS4 SMIL Language Profile	4.0.0	4.1.0
09-2001	13	SP-010457	002		Clarification of H.263 baseline settings	4.0.0	4.1.0
09-2001	13	SP-010457	003	2	Updates to references	4.0.0	4.1.0
09-2001	13	SP-010457	004	1	Corrections to Annex A	4.0.0	4.1.0
09-2001	13	SP-010457	005	1	Clarifications to chapter 7	4.0.0	4.1.0
09-2001	13	SP-010457	006	1	Clarification of the use of XHTML Basic	4.0.0	4.1.0

Change history for TSG-SA4 PSM SWG internal working draft

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2001-06-07				0.0.1	First draft for Release 5 during SA4#17.	TS26234-400	S4-010404
2001-08-29				0.0.2	Editorial changes before SA4#18	S4-010404	S4-010484
2001-09-05				0.1.0	Added text about PNG, capability exchange, Buffering verification and RTP usage model after decision in PSM SWG SA4#18	S4-010484	S4-010515R
2001-10-22				0.2.0	New version prepared with V4.1.0 as base. Changes according to comments received at SA4#18 plenary included. Corrections to sections concerning capability exchange made.	S4-010515R	S4-AHP038
2001-10-30				0.3.0	New SDP attributes to signal pre-decoder buffer settings and the default size of the buffer changed.	S4-AHP038	S4-AHP069
2001-10-31				0.4.0	SVG added as vector graphics format.	S4-AHP069	S4-AHP070
2001-11-28				0.4.1	Editorial updates before SA#19, in preparation for presentation to SA.	S4-AHP070	S4-010614
2001-12-05				0.5.0	Changes about decisions regarding SP-MIDI included during SA4#19.	S4-010614	S4-010651
2001-12-06				0.5.1	Editorial modifications after review at SA4#19	S4-010651	S4-010684
2001-12-07				0.5.1	The wrong version was included in S4-010684 and this document replaces v0.5.1in S4-010684. It also includes updates relating to comments received during the presentation of S4-010651 and S4-010684 SA4#19.	S4-010684	S4-010696