ETSI TS 126 234 V4.3.0 (2002-03)

Technical Specification

Universal Mobile Telecommunications System (UMTS);
End-to-end transparent streaming service;
Protocols and codecs
(3GPP TS 26.234 version 4.3.0 Release 4)

Reference

RTS/TSGS-0426234Uv4R3

Keywords

UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

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 two 3G TSs; 3GPP TS 26.233 [2] and the present document. The first TS 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, scene description, media transport and media encapsulations are specified. Codecs for speech, audio, video, still images, bitmap 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] (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] IETF RFC 3267: " RTP payload format and file storage format for the Adaptive Multi-Rate (AMR) Adaptive Multi-Rate Wideband (AMR-WB) audio codecs ", March 2002.

[12] void

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

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

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

[16] IETF RFC 3236: " The 'application/xhtml+xml' Media Type ", Baker M. and Stark P., January 2002.

[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] ITU-T Recommendation G.722.2 (2002) Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB).

[21] ISO/IEC 14496-3:2001, "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 14496-2:2001, "Information technology -- Coding of audio-visual objects -- Part 2: Visual".

[25] ISO/IEC 14496-2:2001/Amd 2:2002, "Streaming video profile".

[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 (2001): "Information technology - Coding of audio-visual objects - Part 1: Systems".

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

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

[37] (void)

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

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

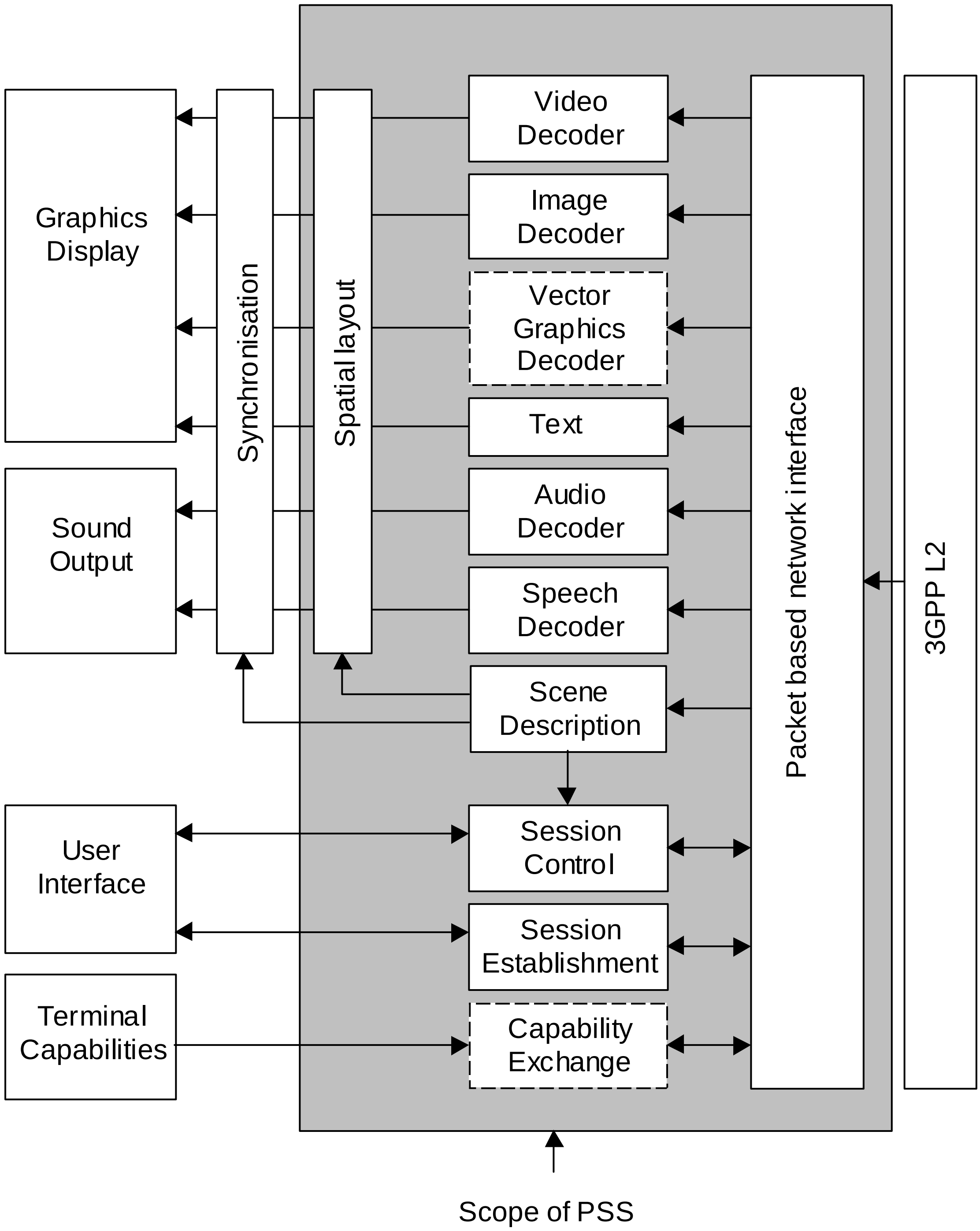
scene description: description of the spatial layout and temporal behaviour of a presentation, it can also contain hyperlinks

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
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
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SDP	Session Description Protocol
SMIL	Synchronised Multimedia Integration Language
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

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, bitmap graphics, text, 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.

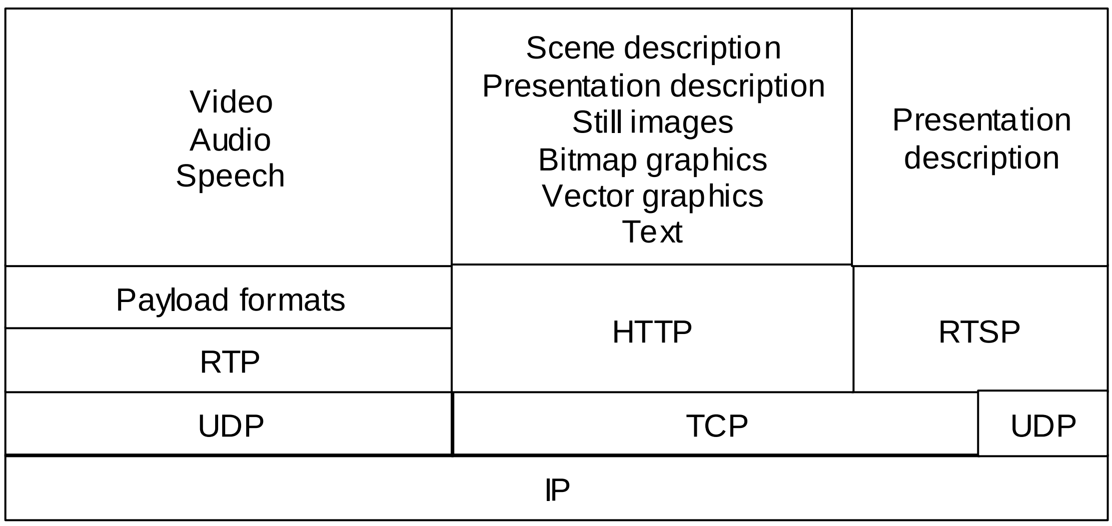


Figure 2: Overview of the protocol stack

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

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

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. For RTP based applications, AS gives the RTP "session bandwidth" (including UDP/IP overhead) as defined in section 6.2 of [9].

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

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 [11];
- 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 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];
- XHTML (see clause 7.8) MIME media type as defined in [16].

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

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]. A PSS client is not required to support multi-channel sessions;
- AMR wide band speech codec (see clause 7.2) RTP payload format according to [11]. A PSS client is not required to support multi-channel sessions;
- 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].

NOTE: The payload format RFC 3016 for MPEG-4 AAC specify that the audio streams shall be formatted by the LATM (Low-overhead MPEG-4 Audio Transport Multiplex) tool [21]. It should be noted that the references for the LATM format in the RFC 3016 [13] point to an older version of the LATM format than included in [21]. In [21] a corrigendum to the LATM tool is included. This corrigendum includes changes to the LATM format making implementations using the corrigendum incompatible with implementations not using it. To avoid future interoperability problems, implementations of PSS client and servers supporting AAC shall follow the changes to the LATM format included in [21].

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].

7 Codecs

7.1 General

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

7.2 Speech

The AMR decoder shall be supported for narrow-band speech [18]. The AMR wideband speech decoder [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 decoder [21] should be supported. The maximum sampling rate to be supported by the decoder is 48 kHz. The channel configurations to be supported are mono (1/0) and stereo (2/0). In addition, the MPEG-4 AAC Long Term Prediction object type decoder may be supported.

7.4 Video

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

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

These two video decoders are optional to implement.

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 decoders in the future (from both the ITU-T and ISO MPEG).

7.5 Still images

ISO/IEC JPEG [26] together with JFIF [27] decoders shall be supported. The support for ISO/IEC JPEG only apply to the following two modes:

- baseline DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF0' in [26];
- progressive DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF2' [26].

7.6 Bitmap graphics

The following bitmap graphics decoders should be supported:

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

7.7 Vector graphics

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

7.8 Text

The text decoder 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 coding formats 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.

9.2.5 Interpretation of MPEG-4 file format

All index numbers used in MPEG-4 file format start with the value one rather than zero, in particular “first-chunk” in Sample to chunk atom, “sample-number” in Sync sample atom and “shadowed-sample-number”, “sync-sample-number” in Shadow sync sample atom.

Annex A (informative): Protocols

A.1 SDP

This clause gives some background information on SDP for PSS clients.

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

Table A.1: Overview of fields in SDP for PSS clients

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
One or more Time Descriptions (See below)				
Z	Time zone adjustments		O	O
K	Encryption key		O	O
A	Session attributes	control	O	R
		range	O	R
One or more Media Descriptions (See below)				
Time Description				
T	Time the session is active		R	R
R	Repeat times		O	O
Media Description				
M	Media name and transport address		R	R
I	Media title		O	O
C	Connection information		R	R
B	Bandwidth information	AS	O	R
K	Encryption Key		O	O
A	Attribute Lines	control	O	R
		range	O	R
		fmtp	O	R
		rtpmap	O	R
Note 1: R = Required, O = Optional Note 2: The "c" type is only required on the session level if not present on the media level. Note 3: The "c" type is only required on the media level if not present on the session level. Note 4: According to RFC 2327, either an 'e' or 'p' field must be present in the SDP description. On the other hand, both fields will be made optional in the future release of SDP. So, for the sake of robustness and maximum interoperability, either an 'e' or 'p' field shall be present during the server's SDP file creation, but the client should also be ready to receive SDP content that does not have neither 'e' nor 'p' fields.				

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

EXAMPLE:

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

A.2 RTSP

A.2.1 General

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

EXAMPLE:

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

RTSP/1.0 200 OK
CSeq: 1
Content-Type: application/sdp
Content-Length: 435
v=0
o=- 950814089 950814089 IN IP4 144.132.134.67
s=Example of aggregate control of AMR speech and H.263 video
e=foo@bar.com

c=IN IP4 192.168.30.29
b=AS:77
t=0 0
a=range:npt=0-59.3478
a=control:*
m=audio 0 RTP/AVP 97
b=AS:13
a=rtpmap:97 AMR/8000
a=fmtp:97 mode-set=0,2,5,7; maxptime=200
a=control:streamID=0
m=video 0 RTP/AVP 98
b=AS:64
a=rtpmap:98 H263-2000/90000
a=fmtp:98 profile=3;level=10
a=control: streamID=1

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

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

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

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

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

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

NOTE: Headers can be folded onto multiple lines if the continuation line begins with a space or horizontal tab. For more information, see RFC2616 [17].

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

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

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

RTSP/1.0 200 OK
CSeq: 5
Session: dfhyrio90llk

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

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

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

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

A.2.2 Implementation guidelines

A.2.2.1 Usage of persistent TCP

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

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

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

A.2.2.2 Detecting link aliveness

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

1. RTCP
2. OPTIONS method with Session header field

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

A.3 RTP

A.3.1 General

Void.

A.3.2 Implementation guidelines

A.3.2.1 Maximum RTP packet size

The RFC 1889 (RTP) [9] does not impose a maximum size on RTP packets. However, when RTP packets are sent over the radio link of a 3GPP PSS system there is an advantage in limiting the maximum size of RTP packets.

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

For these reasons it is recommended that the maximum size of RTP packets should be limited is size taking into account the wireless link. This will decrease the RTP packet loss rate particularly for RLC in UM. For RLC in AM the delay jitter will be reduced permitting the client to use a smaller receiving buffer. It should also be noted that too small RTP packets could result in too much overhead if IP/UDP/RTP header compression is not applied or unnecessary load at the streaming server.

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

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

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

The RFC 2326 (RTSP) [5] specifies that both RTP sequence numbers and RTP timestamps must be continuous and monotonic across jumps of NPT. Thus when a server receives a request for a skip of the media that causes a jump of NPT, it shall specify RTP sequence numbers and RTP timestamps continuously and monotonically across the skip of the media to conform to the RTSP specification. Also, the server may respond with "seq" in the RTP-Info field if this parameter is known at the time of issuing the response.

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.

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

AMR and AMR-WB data is stored in the stream according to the AMR and AMR-WB storage format for single channel header of Annex E [11], without the AMR magic numbers.

D.2 Sample Description atom

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

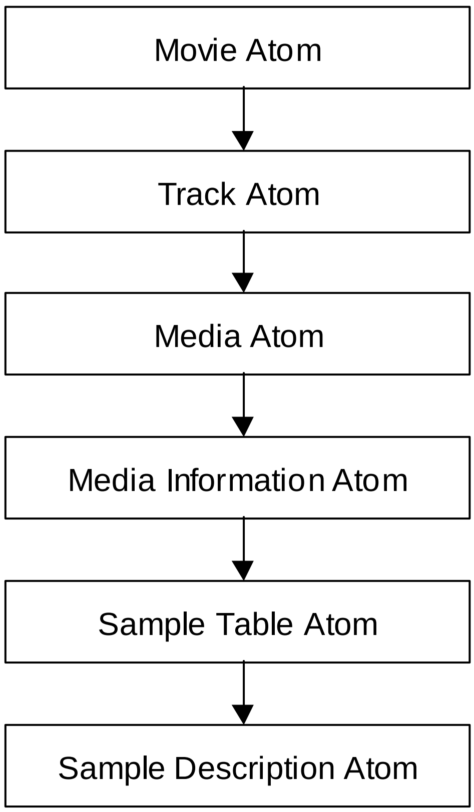


Figure D.1: MP4 Atom Structure Hierarchy

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

Atoms. The Sample DescriptionEntry Atoms for AMR and AMR-WB shall be AMRSampleEntry, and for H.263 shall be H263SampleEntry, respectively.

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

SampleDescriptionEntry ::= VisualSampleEntry |

AudioSampleEntry |

HintSampleEntry |

MpegSampleEntry

H263SampleEntry |

AMRSampleEntry

Table D.1: SampleDescriptionEntry fields

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

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

D.3 VisualSampleEntry atom

The VisualSampleEntry Atom is defined as follows:

VisualSampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_16

Width

Height

Reserved_4

Reserved_4

Reserved_4

Reserved_2

Reserved_32

Reserved_2

Reserved_2

ESDAtom

Table D.2: VisualSampleEntry fields

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		'mp4v'
Reserved_6	Unsigned int(8) [6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference Atoms.	
Reserved_16	Const unsigned int(32) [4]		0
Width	Unsigned int(16)	Maximum width, in pixels of the stream	
Height	Unsigned int(16)	Maximum height, in pixels of the stream	
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0
Reserved_2	Const unsigned int(16)		1
Reserved_32	Const unsigned int(8) [32]		0
Reserved_2	Const unsigned int(16)		24
Reserved_2	Const int(16)		-1
ESDAtom		Atom containing an elementary stream descriptor for this stream.	

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

This version of the VisualSampleEntry, with explicit width and height, shall be used for MPEG-4 video streams conformant to this specification.

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

D.4 AudioSampleEntry atom

AudioSampleEntryAtom is defined as follows:

AudioSampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_8

Reserved_2

Reserved_2

Reserved_4

TimeScale

Reserved_2

ESDAtom

Table D.3: AudioSampleEntry fields

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

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

D.5 AMRSampleEntry atom

For narrow-band AMR, the atom type of the AMRSampleEntry Atom shall be 'samr'. For AMR wide-band (AMR-WB), the atom type of the AMRSampleEntry Atom shall be 'sawb'.

The AMRSampleEntry Atom is defined as follows:

AMRSampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_8

Reserved_2

Reserved_2

Reserved_4

TimeScale

Reserved_2

AMRSpecificAtom

Table D.4: AMRSampleEntry fields

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

If one compares the AudioSampleEntry Atom - AMRSampleEntry Atom the main difference is in the replacement of the ESDAtom, which is specific to MPEG-4 systems, with an atom suitable for AMR and AMR-WB. The **AMRSpecificAtom** field structure is described in clause D.7.

D.6 H263SampleEntry atom

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

The H263SampleEntry Atom is defined as follows:

H263SampleEntry ::= AtomHeader

Reserved_6

Data-reference-index

Reserved_16

Width

Height

Reserved_4

Reserved_4

Reserved_4

Reserved_2

Reserved_32

Reserved_2

Reserved_2

H263SpecificAtom

Table D.5: H263SampleEntry fields

Field	Type	Details	Value
AtomHeader.Size	Unsigned int(32)		
AtomHeader.Type	Unsigned int(32)		's263'
Reserved_6	Unsigned int(8) [6]		0
Data-reference-index	Unsigned int(16)	Index to a data reference that to use to retrieve the sample data. Data references are stored in data reference Atoms.	
Reserved_16	Const unsigned int(32) [4]		0
Width	Unsigned int(16)	Maximum width, in pixels of the stream	
Height	Unsigned int(16)	Maximum height, in pixels of the stream	
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0x00480000
Reserved_4	Const unsigned int(32)		0
Reserved_2	Const unsigned int(16)		1
Reserved_32	Const unsigned int(8) [32]		0
Reserved_2	Const unsigned int(16)		24
Reserved_2	Const int(16)		-1
H263SpecificAtom		Information specific to the H.263 decoder.	

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

D.7 AMRSpecificAtom field for AMRSampleEntry atom

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

Table D.6: The AMRSpecificAtom fields for AMRSampleEntry

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

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

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

The AMRDecSpecStruc is defined as follows:

struct **AMRDecSpecStruc**{

Unsigned int (32) **vendor**

Unsigned int (8) **decoder_version**

Unsigned int (16) **mode_set**

Unsigned int (8) **mode_change_period**

Unsigned int (8) **frames_per_sample**

}

The definitions of AMRDecSpecStruc members are as follows:

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

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

mode_set: the active codec modes. Each bit of the mode_set parameter corresponds to one mode. The bit index of the mode is calculated according to the 4 bit FT field of the AMR or AMR-WB frame structure. The mode_set bit structure is as follows: (B15xxxxxxB8B7xxxxxxB0) where B0 (Least Significant Bit) corresponds to Mode 0, and B8 corresponds to Mode 8.

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

The mapping of existing AMR-WB modes to FT is given in Table E.1-a in [20]. A value of 0x83FF means all modes and comfort noise frames are possibly present in an AMR-WB stream.

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

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

if (mode_change_period < frames_per_sample)

frames_per_sample = k x (mode_change_period)

else if (mode_change_period > frames_per_sample)

mode_change_period = k x (frames_per_sample)

where k : integer [2, …]

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

frames_per_sample: defines the number of frames to be considered as 'one sample' inside the MP4 file. This number shall be greater than 0 and less than 16. A value of 1 means each frame is treated as one sample. A value of 10 means that 10 frames (of duration 20 msec each) are put together and treated as one sample. It must be noted that, in this case, one sample duration is 20 (msec/frame) x 10 (frame) = 200 msec. For the last sample of the stream, the number of frames can be smaller than frames_per_sample, if the number of remaining frames is smaller than frames_per_sample.

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

D.8 H263SpecificAtom field for H263SampleEntry atom

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

The H263SpecificAtom for H263 is composed of the following fields.

Table D.7: The H263SpecificAtom fields H263SampleEntry

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

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

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

H263DecSpecStruc is defined as follows:

struct **H263DecSpecStruc**{

Unsigned int (32) **vendor**

Unsigned int (8) **decoder_version**

Unsigned int (8) **H263_Level**

Unsigned int (8) **H263_Profile**

}

The definitions of H263DecSpecStruc members are as follows:

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

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

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

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

D.9 File Identification

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

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

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

Table D.8: The File-Type atom

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

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

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

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

Annex E (normative): RTP payload format and file storage format for AMR and AMR-WB audio

The AMR and AMR-WB speech codec RTP payload, storage format and MIME type registration are specified in [11].

Annex F (informative): Change history

Change history							
Date	**TSG SA#**	**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
12-2001	14	SP-010703	007		Correction of SDP Usage	4.1.0	4.2.0
12-2001	14	SP-010703	008	1	Implementation guidelines for RTSP and RTP	4.1.0	4.2.0
12-2001	14	SP-010703	009		Correction to media type decoder support in the PSS client	4.1.0	4.2.0
12-2001	14	SP-010703	010		Amendments to file format support for 26.234 release 4	4.1.0	4.2.0
03-2002	15	SP-020087	011		Specification of missing limit for number of AMR Frames per Sample	4.2.0	4.3.0
03-2002	15	SP-020087	013	2	Removing of the reference to TS 26.235	4.2.0	4.3.0
03-2002	15	SP-020087	014		Correction to the reference for the XHTML MIME media type	4.2.0	4.3.0
03-2002	15	SP-020087	015	1	Correction to MPEG-4 references	4.2.0	4.3.0
03-2002	15	SP-020087	018	1	Correction to the width field of H263SampleEntry Atom in Section D.6	4.2.0	4.3.0
03-2002	15	SP-020087	019		Correction to the definition of "b=AS"	4.2.0	4.3.0
03-2002	15	SP-020087	020		Clarification of the index number's range in the referred MP4 file format	4.2.0	4.3.0
03-2002	15	SP-020087	021		Correction of SDP attribute 'C='	4.2.0	4.3.0
03-2002	15	SP-020173	023		References to "3GPP AMR-WB codec" replaced by "ITU-T Rec. G.722.2" and "RFC 3267"	4.2.0	4.3.0

History

Document history		
V4.0.0	March 2001	Publication
V4.1.0	September 2001	Publication
V4.2.0	December 2001	Publication
V4.3.0	March 2002	Publication