

Network Working Group
Request for Comments: 4281
Category: Standards Track

R. Gellens
Qualcomm
D. Singer
Apple
P. Frojdh
Ericsson
November 2005

The Codecs Parameter for "Bucket" Media Types

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

Several MIME type/subtype combinations exist that can contain different media formats. A receiving agent thus needs to examine the details of such media content to determine if the specific elements can be rendered given an available set of codecs. Especially when the end system has limited resources, or the connection to the end system has limited bandwidth, it would be helpful to know from the Content-Type alone if the content can be rendered.

This document adds a new parameter, "codecs", to various type/subtype combinations to allow for unambiguous specification of the codecs indicated by the media formats contained within.

By labeling content with the specific codecs indicated to render the contained media, receiving systems can determine if the codecs are supported by the end system, and if not, can take appropriate action (such as rejecting the content, sending notification of the situation, transcoding the content to a supported type, fetching and installing the required codecs, further inspection to determine if it will be sufficient to support a subset of the indicated codecs, etc.)

Table of Contents

1. Introduction	2
2. Conventions Used in This Document	4
3. The Codecs Parameter	4
3.1. Generic Syntax	5
3.2. ISO File Format Name Space	7
3.3. ISO Syntax	8
4. Use in Additional Media Types	8
5. Examples	9
6. Additional Media Feature Details	9
7. IANA Considerations	9
8. Security Considerations	9
9. Acknowledgements	10
10. Normative References	10
11. Informative References	10

1. Introduction

One of the original motivations for MIME is the ability to identify the specific media type of a message part. However, due to various factors, it is not always possible from looking at the MIME type and subtype to know which specific media formats are contained in the body part, or which codecs are indicated in order to render the content.

There are several media type/subtypes (either currently registered or deployed with registration pending) that contain codecs chosen from a set. It is currently necessary to examine each media element in order to determine the codecs required to render the content. For example, video/3gpp may contain any of the video formats H.263 Profile 0, H.263 Profile 3, H.264, MPEG-4 Simple Profile, and/or any of the audio formats Adaptive Multi Rate (AMR), Adaptive Multi Rate - WideBand (AMR-WB), Extended AMR-WB, Advanced Audio Coding (AAC), or Enhanced aacPlus, as specified in [3GPP-Formats].

In some cases, the specific codecs can be determined by examining the header information of the media content. While this isn't as bad as examining the entire content, it still requires specialized knowledge of each format and is resource consumptive.

This ambiguity can be a problem for various clients and servers. It presents a significant burden to Multimedia Messaging (MMS) servers, which must examine the media sent in each message in order to determine which codecs are required to render the content. Only then can such a server determine if the content requires transcoding or specialized handling prior to being transmitted to the handset.

Additionally, it presents a challenge to smart clients on devices with constrained memory, processing power, or transmission bandwidth (such as cellular telephones and PDAs). Such clients often need to determine in advance if they are currently capable of rendering the content contained in an MMS or email message.

Current ambiguity:

- o audio/3gpp can contain AMR, AAC, AMR-WB, Extended AMR-WB, or Enhanced aacPlus contents as specified in [3GPP-Formats].
- o audio/3gpp2 can contain AMR, AAC, 13K (as per [13k]), Enhanced Variable Rate Codec (EVRC), Selectable Mode Vocoder (SMV), or VMR-WB, as specified in [3GPP2-Formats].
- o video/3gpp can contain H.263 Profile 0, H.263 Profile 3, H.264, MPEG-4 Simple Profile, and/or AMR, AMR-WB, Extended AMR-WB, AAC, or Enhanced aacPlus, as specified in [3GPP-Formats].
- o video/3gpp2 can contain H.263 Profile 0, H.263 Profile 3, H.264, MPEG-4 Simple Profile, and/or AMR, AAC, 13K (as per [13k]), EVRC, SMV, or VMR-WB, as specified in [3GPP2-Formats].

Note that there are additional media types that are ambiguous, but are outside the scope of this document, including:

- o video/mpeg4-generic, which can contain anything allowed by the MPEG-4 specification, or any codec registered with the MP4 registration authority [MP4-Reg];
- o video/quicktime, which can contain anything for which there is a QuickTime codec component; since QuickTime is extensible, this is not limited to the codecs that are or have been shipped by Apple Computer.

With each "bucket" type, a receiving agent only knows that it has a container format. It doesn't even know whether content labeled video/3gpp or video/3gpp2 contains video; it might be audio only, audio and video, or video only.

A solution that permits a receiving agent to determine the specific codecs required to render media content would help provide efficient and scalable servers, especially for Multimedia Messaging (MMS), and aid the growth of multimedia services in wireless networks.

2. Conventions Used in This Document

The key words "REQUIRED", "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS].

The syntax in this document uses the BNF rules specified in [MIME-Format] and [MIME-Coding].

3. The Codecs Parameter

This document adds a parameter to allow unambiguous specification of all codecs indicated to render the content in the MIME part. This parameter is optional in all current types to which it is added. Future types that contain ambiguity are strongly encouraged to include this parameter.

Media types:

- audio/3gpp,
- audio/3gpp2,
- video/3gpp,
- video/3gpp2,

Parameter name:
Codecs

Parameter value: A single value, or a comma-separated list of values identifying the codec(s) indicated to render the content in the body part.

Each value consists of one or more dot-separated elements. The name space for the first element is determined by the MIME type. The name space for each subsequent element is determined by the preceding element.

Note that, per [MIME-Format], some characters (including the comma used to separate multiple values) require that the entire parameter value be enclosed in quotes.

An element MAY include an octet that must be encoded in order to comply with [MIME-Format]. In this case, [MIME-Coding] is used: an asterisk ("*") is placed at the end of the parameter name (becoming "codecs*" instead of "codecs"), the parameter value usually starts with two single quote ("'") characters (indicating that neither character set nor language is specified), and each octet that requires encoding is represented as a percent sign ("%") followed by two hexadecimal digits. Note that, when the [MIME-Coding] form is used, the percent

sign, asterisk, and single quote characters have special meaning and so must themselves be encoded.

Examples of Generic Syntax:

```
codecs=a.bb.ccc.d
codecs="a.bb.ccc.d, e.fff"
codecs*='''fo%2e
codecs*='''%25%20xz, gork"
```

When the Codecs parameter is used, it MUST contain all codecs indicated by the content present in the body part. The Codecs parameter MUST NOT include any codecs that are not indicated by any media elements in the body part.

In some cases, not all indicated codecs are absolutely required in order to render the content. Therefore, when a receiver does not support all listed codecs, special handling MAY be required. For example, the media element(s) MAY need to be examined in order to determine if an unsupported codec is actually required (e.g., there may be alternative tracks (such as English and Spanish audio), there may be timed text that can be dropped, etc.)

NOTE: Although the parameter value MUST be complete and accurate in 'breadth' (that is, it MUST report all four-character codes used in all tracks for ISO-family files, for example) systems MUST NOT rely on it being complete in 'depth'. If the hierarchical rules for a given code (e.g., 'qvxy') were written after a server was implemented, for example, that server will not know what elements to place after 'qvxy'.

If a receiver encounters a body part whose Codecs parameter contains codecs that are not indicated by any media elements, then the receiver SHOULD process the body part by discarding the information in the Codecs parameter.

If a receiver encounters a body part whose Codecs parameter does not contain all codecs indicated by the media elements, then the receiver MAY process the body part by discarding the information in the Codecs parameter.

3.1. Generic Syntax

The Codecs parameter takes either of two forms. The first form is used when the value does not contain any octets that require encoding. The second form uses [MIME-Coding] to allow arbitrary octets to be encoded. With either form, quotes allow for commas and other characters in <tspecials> (quotes MAY be used even when not required).

This BNF uses the rules specified in [MIME-Format] and [MIME-Coding]. Implementations MUST NOT add CFWS between the tokens except after ",", ".".

```
codecs      := cod-simple / cod-fancy
cod-simple  := "codecs" "=" unencodedv
unencodedv  := id-simple / simp-list
simp-list   := DQUOTE id-simple *( "," id-simple ) DQUOTE
id-simple   := element
             ; "." reserved as hierarchy delimiter
element     := 1*octet-sim
octet-sim   := <any TOKEN character>
             ; <TOKEN> defined in [MIME-Format]
             ;
             ; Within a Codecs parameter value, "." is reserved
             ; as a hierarchy delimiter
cod-fancy   := "codecs*" ":@" encodedv
encodedv     := fancy-sing / fancy-list
fancy-sing   := [charset] "'" [language] "'" id-encoded
             ; Parsers MAY ignore <language>
             ; Parsers MAY support only US-ASCII and UTF-8
fancy-list   := DQUOTE [charset] "'" [language] "'" id-list DQUOTE
             ; Parsers MAY ignore <language>
             ; Parsers MAY support only US-ASCII and UTF-8
id-list      := id-encoded *( "," id-encoded )
id-encoded   := encoded-elm *( "." encoded-elm )
             ; "." reserved as hierarchy delimiter
encoded-elm  := 1*octet-fancy
octet-fancy  := ext-octet / attribute-char
             ; <ext-octet> and <attribute-char> defined in
             ; [MIME-Coding]
DQUOTE       := %x22 ; " (double quote)
```

Initial name space: This document only defines values for files in the ISO Base Media File Format family. Other file formats may also define codec naming.

3.2. ISO File Format Name Space

For the ISO Base Media File Format, the first element of a Codecs parameter value is a sample description entry four-character code as registered by the MP4 Registration Authority [MP4-Reg]. Values are case sensitive.

Note that there are potentially multiple tracks in a file, each potentially carrying multiple sample entries (some but not all uses of the ISO File Format restrict the number of sample entries in a track to one).

When the first element of a value is 'mp4a' (indicating some kind of MPEG-4 audio) or 'mp4v' (indicating some kind of MPEG-4 part-2 video), the second element is the hexadecimal representation of the MP4 Registration Authority ObjectTypeIndication (OTI), as specified in [MP4-Reg] and [MP41] (including amendments). Note that [MP4-Reg] uses a leading "0x" with these values, which is omitted here and hence implied.

One of the OTI values for 'mp4a' is 40 (identifying MPEG-4 audio). For this value, the third element identifies the audio ObjectTypeIndication (OTI) as defined in [MP4A] (including amendments), expressed as a decimal number.

For example, AAC low complexity has the value 2, so a complete string for AAC-LC would be "mp4a.40.2".

One of the OTI values for 'mp4v' is 20 (identifying MPEG-4 part-2 video). For this value, the third element identifies the video ProfileLevelIndication as defined in [MP4V] (including amendments), expressed as a decimal number.

For example, MPEG-4 Visual Simple Profile Level 0 has the value 9, so a complete string for MPEG-4 Visual Simple Profile Level 0 would be "mp4v.20.9".

3.3. ISO Syntax

```

id-simple    :=/ id-iso
id-encoded   :=/ id-iso
id-iso       := iso-gen / iso-mpega / iso-mpegv
iso-gen      := cpid *( element / encoded-elm )
               ; <element> used with <codecs-simple>
               ; <encoded-elm> used with <codecs-fancy>
               ;
               ; Note that the BNF permits "." within <element>
               ; and <encoded-elm> but "." is reserved as the
               ; hierarchy delimiter

iso-mpega    := mp4a "." oti [ "." aud-oti ]
iso-mpegv    := mp4v "." oti [ "." vid-pli ]

cpid         := 4(octet-simple / octet-fancy)
               ; <octet-simple> used with <codecs-simple>
               ; <octet-fancy> used with <codecs-fancy>

mp4a         := %x6d.70.34.61 ; 'mp4a'

oti          := 2(DIGIT / "A" / "B" / "C" / "D" / "E" / "F")
               ; leading "0x" omitted

aud-oti      := 1*DIGIT

mp4v         := %x6d.70.34.76 ; 'mp4v'
vid-pli      := 1*DIGIT

```

4. Use in Additional Media Types

This parameter MAY be specified for use with additional MIME media types.

For ISO file formats where the name space as defined here is sufficient, all that needs to be done is to update the media type registration to specify the Codecs parameter with a reference to this document. For existing media types, it is generally advisable for the parameter to be optional; for new media types, the parameter MAY be optional or required, as appropriate.

For ISO file formats where the name space as defined here needs to be expanded, a new document MAY update this one by specifying the additional detail.

For non-ISO formats, a new document MAY update this one by specifying the name space for the media type(s).

5. Examples

Content-Type: video/3gpp2; codecs="sevc, s263"
(EVRc audio plus H.263 video)

Content-Type: audio/3gpp; codecs=samr
(AMR audio)

Content-Type: video/3gpp; codecs="s263, samr"
(H.263 video plus AMR audio)

Content-Type: audio/3gpp2; codecs=mp4a.E1
(13k audio)

Content-Type: video/3gpp2; codecs="mp4v.20.9, mp4a.E1"
(MPEG-4 Visual Simple Profile Level 0 plus 13K voice)

Note: OTI value 20 ("0x20" in [MP4-Reg]) says "Includes associated Amendment(s) and Corrigendum(a). The actual object types are defined in [MP4V] and are conveyed in the DecoderSpecificInfo as specified in [MP4V], Annex K." (references adjusted).

6. Additional Media Feature Details

It is sometimes helpful to provide additional details for a media element (e.g., the number of X and Y pixels, the color depth, etc.). These details are sometimes called "media features" or "media characteristics".

When such additional features are included, the [Content-Features] header provides a handy way to do so.

7. IANA Considerations

The IANA has added "codecs" as an optional parameter to the media types listed in Section 3, with a reference to this document.

8. Security Considerations

The Codecs parameter itself does not alter the security considerations of any of the media types with which it is used. Each audio and video media type has its own set of security considerations that continue to apply, regardless of the use of the Codecs parameter.

An incorrect Codecs parameter might cause media content to be received by a device that is not capable of rendering it, or might cause media content to not be sent to a device that is capable of

receiving it. An incorrect Codecs parameter is therefore capable of some types of denial-of-service attacks. However, this is most likely to arise by accident, as an attacker capable of altering media data in transit could cause more harm by altering the media format itself, or even the content type header, rather than just the Codecs parameter of the content type header.

9. Acknowledgements

Harinath Garudadri provided a great deal of help, which is very much appreciated. Mary Barnes and Bruce Lilly provided detailed and helpful comments. Reviews and comments by Sam Hartman, Russ Housley, and Bert Wijnen were much appreciated. Chris Newman carefully reviewed and improved the BNF.

10. Normative References

- [Content-Features] Klyne, G., "Indicating Media Features for MIME Content", RFC 2912, September 2000.
- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [MIME-Coding] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.
- [MIME-Format] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [Media-Features] Holtman, K., Mutz, A., and T. Hardie, "Media Feature Tag Registration Procedure", BCP 31, RFC 2506, March 1999.
- [MP4-Reg] MP4REG, The MPEG-4 Registration Authority, URL: <<http://www.mp4ra.org>>.

11. Informative References

- [13k] Gellens, R. and H. Garudadri, "The QCP File Format and Media Types for Speech Data", RFC 3625, September 2003.

- [3GPP-Formats] TS 26.244, Third Generation Partnership Project (3GPP), "Transparent End-to-End Packet Switched Streaming Service; 3GPP file format (3GP)", URL: <<http://www.3gpp.org/ftp/Specs/html-info/26244.htm>>.
- [3GPP2-Formats] Third Generation Partnership Project 2, "3GPP2 File Formats for Multimedia Service", URL: <http://www.3gpp2.org/Public_html/specs/C.S0050-0_v1.0_121503.pdf>.
- [MP41] ISO/IEC 14496-1:2004, "Information technology-- Coding of audio-visual objects--Part 1: Systems".
- [MP4A] ISO/IEC 14496-3:2001, "Information technology-- Coding of audio-visual objects--Part 3: Audio".
- [MP4V] ISO/IEC 14496-2:2004, "Information technology-- Coding of audio-visual objects--Part 2: Visual".

Authors' Addresses

Randall Gellens
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121
USA

Email: randy@qualcomm.com

David Singer
Apple Computer, Inc.
One Infinite Loop, MS:302-3MT
Cupertino CA 95014
USA

Email: singer@apple.com
Phone: +1 408 974 3162

Per Frojdh
Ericsson Research
Multimedia Technologies
SE-164 80 Stockholm, SWEDEN

Email: Per.Frojdh@ericsson.com
Phone: +46 8 7190000

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

