



Hey John,
Well here I am
in Rome.
This is the
Colosseum.



MMS
A new generation of
messaging



MMS Developer's Guide

Contents

Preface	3
Purpose of this Document.....	3
Audience.....	3
Building an MMS Message	4
SMIL	6
Examples.....	6
Tags.....	8
Media Formats	13
Images.....	13
Text.....	13
Audio.....	13
MIME Encapulations	14
Example.....	14
Location & ID.....	15
MMS Example.....	16
Reference Documentations	18
Ericsson.....	18
3G PP.....	18
WAP Forum.....	18
W3C.....	18
IETF.....	18

Preface

Purpose of this Document

This document provides a general introduction to how MMS messages are composed. It does not cover the relaying, delivery, or receiving of MMS messages. It gives particular focus to MMS messages and presentation format that adheres to the conformance agreements between Ericsson and Nokia for their first MMS client implementations. Details of the protocol and MMS unique headers can be found in [R2] SMTP Interface Description For Content Providers.

Audience

This document is aimed at the developer who has some technical familiarity with email messages and mark-up languages (HTML, XML, etc.) and wants to create applications capable of composing MMS messages.

Building an MMS Message

MMS is the Multimedia Messaging System. It is the evolution of SMS, the Short Messaging System, a system for sending and receiving short text-based messages. SMS has proven to be a wildly popular feature of 2G GSM networks. MMS is the next generation upgrade for SMS in 3G networks. However, MMS does not require a 3G network. MMS can function under a 2G or 2.5G network, and handsets and networks that support MMS are expected by the end of 2001.

MMS is different from SMS in many ways. The first and most obvious one is its handling of multimedia objects. MMS also offers more integrated support for different messaging standards such as email. Additionally, MMS offers newer delivery features like receive confirmation, read confirmation, and pre-paid replies.

This document should give you an understanding of how to compose, package and address an MMS message. Understanding how MMS messages are put together and sent requires understanding several existing specifications for messaging. The good thing is, you are probably already familiar with many of these specs. In this document, I will provide an overview of the most important of these different specs and show how they come together in the larger MMS specification.

There are two telephony standards bodies producing specifications relating to MMS messages and how they are composed and sent: the WAP (Wireless Application Protocol) Forum and the 3GPP (3rd Generation Partnership Project). The standards produced by these two bodies in turn use existing specifications from two Internet standards bodies: the W3C (World Wide Web Consortium) and the IETF (Internet Engineering Task Force). The standards from the WAP Forum specify how messages are composed and packaged, and the standards from the 3GPP specify how messages are sent, routed, and received.

The goal of this document is to provide an explanation of all of the disparate standards and to show how they come together in MMS. Additionally, this document tempers those standards with information about how they will actually be implemented—which features will be supported and which will be left out—in the first generation of MMS systems.

An MMS message is composed by a user on her mobile phone, computer, PDA, or any other device with an MMS client. MMS messages can also be automatically generated and sent through software. For example, a user could ask for the day's weather forecast to be sent to her each morning complete with animated maps and audio of the weatherman. The weather service would then generate the maps and audio, assemble them into an MMS message and send them to her MMS address.

The first supported MMS messages should all be thought of as slide shows. They may be slide shows with only one slide, but they are slide shows nonetheless. Each of the slides has its display area divided into different sections, such that the slides themselves are really just frames that hold the content (which is kept separate). At first, there will only be two sections per slide—one for an image and one for text. (It is also acceptable to have either just an image region or just a text region.) The layout and ordering of the slides is specified in a language called SMIL. SMIL is covered in the next subsection.

The contents of the slides—the actual images, text, and audio—are separate pieces that are sent along with the slides. These attached files have to be encoded with a supported format. Supported formats are covered in the subsection after SMIL.

Finally, the slides and their contents have to be packaged into one file that can be sent as a message. The maximum size of the entire packaged message that first generation devices will support is 50 kB. This packaging is covered in the final subsection of this section.

SMIL

SMIL, the Synchronized Multimedia Integration Language, is an XML-based language specified by the W3C. It is used to control the presentation of multimedia elements that need information about both static layout and timing for audio and video. Most commonly, SMIL is used to layout complementary text and images around streaming video presentations viewed with RealPlayer or Windows Media Player. The full set of SMIL tags can specify anything from exact, pixel-perfect on-screen position, to simple animations, to interactive video and audio controls.

For the first rollouts of MMS, a very simple subset of Basic SMIL will be supported. And, all of the exact positioning capabilities of SMIL will be enhanced with a layout flexibility closer to that of HTML.

Example

This is a simple example of the SMIL for an MMS message.

```
<smil>
  <head>
    <meta name="title" content="vacation photos" />
    <meta name="author" content="Danny Wyatt" />
    <layout>
      <root-layout width="160" height="120"/>
      <region id="Image" width="100%"
        height="80" left="0" top="0" />
      <region id="Text" width="100%"
        height="40" left="0" top="80" />
    </layout>
  </head>
  <body>
    <par dur="8s">
      
      <text src="FirstText.txt" region="Text" />
      <audio src="FirstSound.amr"/>
    </par>
    <par dur="7s">
      
      <text src="SecondText.txt" region="Text" />
      <audio src="SecondSound.amr" />
    </par>
  </body>
</smil>
```

As you can see, the SMIL mark-up is very similar to HTML mark-up. The entire message body is enclosed in `<smil></smil>` tags and the message (or document) itself has both `head` and `body` sections.

The `head` section contains information that applies to the entire message. The `title` and `author` meta fields here correspond to the `From` and `Subject` fields of the message. These meta fields are not mandatory. The receiving client must be able to handle a message that contains meta fields, but it does not have to actually read the fields or deal with the information in a meta field.

The `layout` section within the `head` section specifies the master layout for all the slides in the message. The example here is for a terminal whose screen will display the slide in a portrait orientation, where the height is greater than the width. It is in this layout specifying that SMIL for MMS and SMIL used on a PC begin to diverge.

On a PC screen, the SMIL slides would all be displayed exactly 160 pixels wide and 120 pixels tall. The total slide area would be divided into two smaller areas. The `image` region would be 80 pixels tall and always appear above the 40 pixel tall `text` area. On an MMS client however, the screen may not be large enough to accommodate the layout. Or the screen may have a different orientation that could better display the slides with another layout. Under MMS implementations of SMIL, a client is free to reformat the layout in a way best suited to the client's display. It is for this reason that the first implementations of MMS terminals only allow one image region and one text region per slide. A device may choose to replace any incoming layout information with its own fixed layout—one that it uses for all MMS messages regardless of their specified layouts. This does not mean that you can compose MMS messages without `layout` sections, however. You must include them for devices that are capable of handling flexible layouts. This restriction will not only make MMS messages easier to display on a wide variety of devices, it will also make it easier for users to compose MMS messages on devices with limited input abilities.

Within the `body` section are the actual slides in the message. These slides are denoted with the `par`—for parallel—tag. Parallel denotes that all the elements within the tag are to be displayed simultaneously. This is obvious for an MMS message, but in full-fledged SMIL the `par` tag is the counterpart to the `seq` (sequence) tag and there are more nuances to its use. In MMS SMIL, the body is implicitly a `seq`, and is the only `seq` available. The `dur` attribute for each slide is the duration of the slide in the slide show. Again, the receiving client is free to modify or ignore this—replacing duration with a button for the next slide, for example—but it should always be included in the message.

Each slide in turn contains at least two elements: one for the `image` region and one for the `text` region. Two of the slides also contain an audio element that will be played when the slide is viewed. In normal SMIL, the names of the layout regions (`image` and `text` in our MMS message) are just handy names for generic regions that can contain any type of content. In MMS SMIL, however, the `image` region must contain an image element and the `text` region a text element.

Tags

The sample message above contains most of the tags that are safe to use for MMS messages in the first generation of MMS implementations. This section is a more thorough reference of a recommended subset of SMIL for MMS. Beneath each tag are that tag's allowable attributes and child tags.

```
<smil></smil>
```

These tags specify the SMIL mark-up language and must surround the entire message data.

Attributes: none

Children: `head`, `body`

```
<head></head>
```

The head tags contain data that applies to the entire message

Attributes: none

Children: `meta`, `layout`

```
<body></body>
```

The body tags contain the body of the message. They enclose the actual slides.

Attributes: none

Children: `par`

```
<meta />
```

The meta tag allows for meta-information about the message to be put in the messages head.

Attributes:

name the name for the meta-information

content the actual meta-information

```
<layout></layout>
```

The layout tag specifies the master layout for all the slides in a message.

Attributes: none

Children: region

```
<root-layout />
```

The root-layout tag specifies the entire area that the message should fill. The maximum area for which interoperability is guaranteed is 160 x 120 pixels.

Attributes:

width the width of the entire area

height the height of the entire area

Children: none

```
<region />
```

The region tag defines a region within a slide. Currently, there are only two valid regions: one for the image and one with the text.

Attributes:

id The name of the region

<code>top</code>	The topmost edge of the region, in pixels from the upper-left corner of the screen
<code>left</code>	The leftmost edge of the region, in pixels from the upper-left corner of the screen
<code>width</code>	The width, in pixels, of the region
<code>height</code>	The height, in pixels, of the region
<code>fit</code>	How the contents of the region should be changed to fit an area different than that specified. The valid values for <code>fit</code> are listed below. For all values, drawing of the object within the region always begins in the upper left corner of the region. Using the <code>fit</code> attribute in MMS messages is not recommended .
<code>fill:</code>	scales the width and height independently to fill all available space
<code>hidden:</code>	clips the bottom and/or right sides if the object is larger than the region or fills the background if the object is smaller than the region
<code>meet:</code>	scales the object up, preserving its aspect ratio, until either its height or width fits the height or width of the region
<code>slice:</code>	scales the object down, preserving its aspect ratio, until either its height or width fits the height or width of the region.
Children:	none

`<par></par>`

The `par` tag marks one slide in the message.

Attributes:

<code>begin</code>	The absolute beginning time for this slide.
<code>end</code>	The absolute ending time for this slide.
<code>dur</code>	The duration of this slide.
Children:	<code>img</code> , <code>text</code> , <code>audio</code> , <code>ref</code>

``

The `img` tag denotes which image to display as this slide's only image.

Attributes:

`src` the source of the image, required, must be a valid image

`region` should be "image" in the first implementations

`alt` alternative text for the image

Children: none

`<text />`

The `text` tag denotes which text to display as this slide's text copy.

Attributes:

`src` the source of the message text, required, must be a valid text block

`region` should be "text" in the first implementations

`alt` alternative display text (somewhat redundant for the `text` region)

Children: none

`<audio />`

The `audio` tag denotes which sound to play when this slide is viewed.

Attributes:

`src` the source of the audio file, required, must be a valid audio file

`alt` alternative text for the audio

Children: none

`<ref />`

The `ref` tag is a generic media object tag. In normal SMIL it can be used instead of `img`, `audio`, `text`, etc. Using the `ref` tag in MMS messages is **not recommended**.

Attributes:

`src` the source of the object

`alt` alternative text for the object

Children: none

Media Formats

The following are the specific media formats that will be supported in the first generation of MMS systems. The image and audio formats are, of course, binary formats. They will also need to be encoded in a format that can safely make it through text-based SMTP systems. Any internet email compliant encoding scheme is allowed, but BASE64 encoding is recommended.

Eventually, when receiving clients begin to support UAPROF, it is possible that MMS systems will transcode elements from an unsupported format to a supported format—but that should not be relied on.

Images

For the first generation of MMS messages there are very few media formats that are guaranteed to be supported across clients. For images, these are baseline JPEG with JFIF exchange format, GIF87a, GIF89a, and WBMP. The maximum guaranteed image resolution is 160 pixels wide by 120 pixels high, larger images are supported, but need to be converted for the target device.

The browser safe color palette (216 colors) used by Netscape or Internet Explorer is recommended for color image for optimal rendering on multiple devices.

When a device supports both JPEG and GIF, JPEG is a better choice for rendering photographs and GIF is a better choice for line drawings.

Text

The text of the message (but not necessarily the text of the markup) may use us-ascii, utf-8, or utf-16 character encoding. The supported character sets on any client will always be at least all of ISO 8859-1.

Audio

Audio should be encoded as AMR (Adaptive Multi Rate, a codec used for voice in GSM and 3G networks). (Many clients will also support iMelody for ring tones.)

MIME Encapsulation

The SMIL example above and all of its attendant media objects will work fine if a SMIL client requests them from a web server. The locations (`src` attributes) of the media object referred to in the SMIL will be resolved as relative to the base document and then requested from the server. For an MMS message though, we need some way of packaging all these objects together so they can be sent as one unit, but so that their references to one another remain valid.

The solution to this problem comes in several parts, all of them built on MIME. MIME is the Multipart Internet Mail Extensions specification. It is a standard originally developed for including content in email messages in addition to the plain text body of the email. If you've ever sent or received an email attachment, you've used MIME. MIME is used to bundle all of the separate audio, image, and text files together—as well as the base SMIL document. Then, an extension to MIME known as “MIME Encapsulation of Aggregate Documents” is used to let a client know that all of the parts of this message are related to one another and may refer to one another. This standard was originally developed around HTML, but has since been expanded to include any type of document that needs to link in other resources.

Example

Here is the example of MIME encapsulation given in the actual RFC for “MIME Encapsulation of Aggregate Documents.”

```
Content-Type: multipart/related; boundary="boundary-example";
           type="text/html"

--boundary-example

Content-Type: text/html; charset="US-ASCII"

... .. <IMG SRC="fiction1/fiction2"> ... ..
... .. <IMG SRC="cid:97116092811xyz@foo.bar.net"> ... ..

--boundary-example
Content-Type: image/gif
Content-ID: <97116092511xyz@foo.bar.net>
Content-Location: fiction1/fiction2

--boundary-example
Content-Type: image/gif
Content-ID: <97116092811xyz@foo.bar.net>
Content-Location: fiction1/fiction3
```

--boundary-example--

The `Content-Type` at the very top of the message indicates to the receiving client that the separate parts of the message are related and may refer to one another. The `boundary` is something common to all multipart messages and indicates to the client what string will separate each of the parts of the message. The boundary string chosen here is for this example only. It would be a poor choice in a real application since the text "boundary-example" could appear somewhere within the message itself. Normally, the boundary string is a long sequence of random characters with very little likelihood of occurring within a message part.

Between each of the boundaries you see the parts of the message. The first part shows only relevant excerpts from the HTML. The second and third parts omit the actual bodies of the images (more on those below) and just show the information relevant to their aggregation in a multipart/related message.

Location and ID

As you can see, the HTML part can refer to an included image by either its specified `Content-ID` or `Content-Location`. These are both ways of identifying parts of the message uniquely so that other parts of the message can refer to them.

`Content-ID` will always be available for each part of the message. If part of the message wishes to refer to another part by `Content-ID` then the scheme "cid:" is used within the reference to that part. Notice that the actual `Content-ID` headers enclose the part's ID in angle brackets. The way internal message references with the `cid:` scheme are resolved is to remove the "cid:" and enclose the remaining string in "<" and ">" and then match against specified `Content-IDs`. So, `Content-ID` references do not contain angle brackets but do start with "cid:" Actual `Content-IDs` do not start with "cid:" but are enclosed in angle brackets.

The `Content-Location` header is there to make it easier to refer to message parts within a base document before the message is assembled and to allow for things like HTML pages to be packaged without having all of their internal links rewritten to refer to `Content-IDs` instead of URLs. In the example above you can see how the HTML part refers to the image `fiction2` just as if it were a normal image accessed via a normal URL. It is the client reading this HTML, that knows the HTML is part of a `multipart/related` message, that looks first within the message parts for that URL before looking out the network. In fact, the client might never look to the network for resources at all; it might only look for them within the message parts.

MMS Example

For an actual MMS message the header information is slightly different than that for an email. The Content-Type is MMS specific and the other headers (more on those below) are unique to MMS. However, they are all compatible with standard email systems—this is a requirement in the MMS specification.

```
Content-Type: multipart/related;
            start=<foo9836mm@mms.domain.com>;
            boundary="boundary123456789";
            type=application/smil
```

```
--boundaryq123456789
Content-ID: <foo9836mm@mms.domain.com>
Content-Type: application/smil; charset="US-ASCII"
```

```
<smil>
  <head>
    <meta name="title" content="vacation photos" />
    <meta name="author" content="Danny Wyatt" />
    <layout>
      <root-layout width="176" height="216"/>
      <region id="Image" width="176"
        height="144" left="0" top="0" />
      <region id="Text" width="176"
        height="72" left="0" top="144" />
    </layout>
  </head>
  <body>
    <par dur="8s">
      
      <text src="bundled/FirstText.txt" region="Text" />
      <audio src="bundled/FirstSound.amr"/>
    </par>
    <par dur="7s">
      
      <text src="bundled/SecondText.txt" region="Text" />
      <audio src="bundled/SecondSound.amr" />
    </par>
  </body>
</smil>
```

```
--boundary123456789
Content-ID: <foo1345mm@mms.domain.com>
Content-Location: bundled/FirstImage.jpg
Content-Type: image/jpeg
```

[. . .]

```
--boundary123456789-
Content-ID: <foo27845mm@mms.domain.com>
Content-Location: bundled/SecondImage.jpg
Content-Type: image/jpeg
```

[. . .]

```
--boundary123456789
Content-ID: <foo2349mm@mms.domain.com>
Content-Location: bundled/FirstText.txt
Content-Type: text/plain
```

This is the text of the first slide.

```
--boundary123456789
Content-ID: <foo2345mm@mms.domain.com>
Content-Location: bundled/SecondText.txt
Content-Type: text/plain
```

This is the text of the second slide.

```
--boundary123456789
Content-ID: <foo2346mm@mms.domain.com>
Content-Location: bundled/FirstSound.amr
Content-Type: audio/AMR
```

[. . .]

```
--boundary123456789
Content-ID: <foo2374mm@mms.domain.com>
Content-Location: bundled/SecondSound.amr
Content-Type: audio/AMR
```

[. . .]

```
--boundary123456789--
```

I've only included the MIME header information for the aggregated parts (except for the short text parts), but hopefully that is enough to make the structure of the message clear.

The `start` parameter (which is a part of the standard multipart/related MIME type) tells the client which part of the message is the base document that specifies the layout and presentation. The `start` parameter can be omitted, but then the base part (the SMIL part) must be the very first part. The order of the other parts does not matter.

There is one other MMS specific details to notice in the `Content-Type` header: the `start` and `type` parameters are not enclosed in double quotes.

Reference Documents

Ericsson

[R1] MMS Conformance Document

3GPP

[R2] SMTP Interface Description For Content Providers

[R3] TS 22-140: Service Aspects

http://www.3gpp.org/ftp/Specs/2001-03/Rel-4/22_series/22140-410.zip

[R4] TS 23-140: Functional Specification

http://www.3gpp.org/ftp/Specs/2001-03/Rel-4/23_series/23140-420.zip

WAP Forum

(Note: you must click through a license agreement to access these documents.)

[R5] WAP-205: MMS architecture overview

<http://www1.wapforum.org/tech/terms.asp?doc=WAP-205-MMSArchOverview-20010425-p.pdf>

[R6] WAP-206: MMS client transactions

<http://www1.wapforum.org/tech/terms.asp?doc=WAP-206-MMSCTR-20010412-p.pdf>

[R7] WAP-209 (MMS message encapsulation)

<http://www1.wapforum.org/tech/terms.asp?doc=WAP-209-MMSEncapsulation-20010601-a.pdf>

W3C

[R8] SMIL

<http://www.w3.org/AudioVideo/>

IETF

[R9] RFC2387: multipart/related MIME type

<http://www.ietf.org/rfc/rfc2387.txt>

[R10] RFC2557: MIME Aggregation

<http://www.ietf.org/rfc/rfc2557.txt>

Ericsson is shaping the future of Mobile and Broadband Internet communications through its continuous technology leadership. Providing innovative solutions in more than 140 countries, Ericsson is helping to create the most powerful communication companies in the world.

This White Paper is published by:

Ericsson Mobility World USA
55 Broad Street
New York, New York 10004

Phone: +1 212 612-1299
Fax: +1 212-612-1289

www.ericsson.com/mobilityworld

First Edition *October 2001*
EN/LZT 108 5354 R1

This document is published by Ericsson Mobility World USA, without any warranty.

Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and /or equipment may be made by Ericsson Mobility World at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Any hard copies of this document are to be regarded as temporary reference copies only.