

## Acronym Cheat Sheet

<b>CSS</b>	Cascading Style Sheets Style sheets define how content should be rendered (font, color, spacing) in a Web document. Each HTML tag (<BODY>, <P>, ...) can have its own style.	(17-Dec-1996/11-Jan-1999)
<b>CSS2</b>	Cascading Style Sheets, Level 2 Builds on CSS. Media-specific style sheets that can be tailored to visual browsers, text readers, printers, braille devices, etc.	(12-May-1998)
<b>DHTML</b>	Dynamic HTML The mixture of JavaScript, HTML, DOM and CSS for dynamic pages (content created on the fly, navigation effects, pull-down and pop-up menus,...).	
<b>DOM</b>	Document Object Model The roadmap to the internal hierarchy of the elements (window/frame, document, form, form elements) in the document.	(Working Draft, 19-April-2001)
<b>DTD</b>	Document Type Definition The "grammar" for your markup language: the set of elements (tags), their attributes, and legal structure, to mark up a document for a particular application.	
<b>HTML</b>	HyperText Markup Language The standard publishing language of the World Wide Web. An application of SGML.	
<b>ISO</b>	International Organization for Standardization A worldwide federation of national standards bodies, one from each of over 100 countries. US member is ANSI (American National Standards Institute).	
<b>MIME</b>	Multipurpose Internet Mail Extensions MIME (as in MIME content-type) defines a format and framework for representing a wide variety of data types in Internet applications.	
<b>SGML</b>	Standard Generalized Markup Language ISO 8879 standard markup language for authors to describe the structure of their documents. HTML is an application of SGML.	
<b>SMIL</b>	Synchronized Multimedia Integration Language Markup language for multimedia presentations.	
<b>XHTML</b>	HTML with the rigor of XML XHTML 1.0 is a reformulation of HTML 4.01 in XML.	
<b>XML</b>	Extensible Markup Language Simplified version of SGML. Markup for the structure of a document.	(XML 1.0, Second ed. 6-May-2000)
<b>XSL</b>	XML Style sheet Language To view an XML document, need to format/style it. Styling instructions are organized in style sheets such as CSS and XSL. XSL has more advanced styling than CSS: it can transform a document before displaying it.	

## Contents

### §1. Background

- I. **History of Web Presentation**
- II. **W3C: World Wide Web Consortium**

### §2. HTML

- I. **Overview**
- II. **Basic Structure of HTML Document**
- III. **Work Flow**
- IV. **Terms**
- V. **Style Sheets**
- VI. **Examples**
- VII. **Summary**

### §3. XML

- I. **Overview**
- II. **DTD: Document Type Definition**
- III. **Strict Syntax**
- IV. **Terms**
- V. **Namespace**
- VI. **XHTML 1.0**

### §4. SMIL Players

- I. **PC/Macintosh**
- II. **Unix**

## §5. SMIL Syntax

### I. Overview

- A. What is SMIL good for?
- B. Basic Structure of SMIL Document

### II. <head> Section

- A. <layout>
  - 1. Window Layout
    - a) root-layout
    - b) region
  - 2. Layout types
- B. <meta>

### III. <body> Section

- A. Media Object Elements
- B. Text Media Types
  - 1. <text>
    - a) text/plain
    - b) text/html
  - 2. <textstream>
- C. Synchronization
- D. Adapting the Presentation
- E. Hyperlinking

## §6. Reference Sites

- I. Recommendations/Specifications
- II. Tutorials
- III. Maillist
- IV. Authoring Tools

§Appendix-1. SMIL 1.0 Specification

§Appendix-2. MIME types

## Talk Outline

- §1. Background      Short History of the Web
- §2. HTML            Where we come from.  
                         Define terms.  
                         Talk about short-comings.
- §3. XML             Where it's going.  
                         SMIL is XML-based...  
                         Learn the basics & terminology.
- §4. SMIL Players    Free players to watch our presentations.  
                         Choices for PC, Mac, and Unix.
- §5. SMIL Syntax    Markup for multimedia presentations.  
                         Lecture/lab.

## §1. Background

### I. History of Web Presentation

HTML, HyperText Markup Language, is the current language for web page authoring. The term hypertext was coined in 1965 by Ted Nelson (*A File Structure for the Complex, the Changing, and the Indeterminate*. 20th National Conference, New York, Association for Computing Machinery, 1965). The intent of HTML was for hyper linking text documents.

HTML's ease of use, non-proprietary format, and widespread acceptance has caused it to be used for multimedia purposes for which it was not designed. The markup tags are pre-defined, and may not suit your needs, but you try anyway. One common hack is to lay out your images and text by clever use of tables. Another fundamental problem is that web page content is intermixed with its presentation. Style sheets were added to help separate content from presentation. Interactivity requires scripting (JavaScript, CGI), animated gifs, or programming (Java applets, Flash, Director).

XML, Extensible Markup Language, was designed as a standard to take care of these limitations. An author defines the tags needed for the application at hand, and a model that describes every element that can appear in the documents. The focus is on the structure of the document, and the meaning of the content. A separate model, the XSL, specifies how the content shall be rendered. The same content can be presented in different ways (HTML pages, data for other applications, audio) as needed.

A few XML-based standards have been designed for special-purpose communities. Here we'll look at SMIL (pronounced "smile"): Synchronized Multimedia Integration Language.

### II. W3C: World Wide Web Consortium

[www.w3.org](http://www.w3.org)

W3C are the keepers of the standards, founded in 1994 by Tim Berners-Lee, the inventor of the web as we know it today. Through working groups of industry experts, their mission is the development and maintenance of web standards.

Here are just a few of the relevant sites:

<a href="http://www.w3.org/MarkUp">www.w3.org/MarkUp</a>	HTML
<a href="http://www.w3.org/AudioVideo">www.w3.org/AudioVideo</a>	SMIL
<a href="http://www.w3.org/XML">www.w3.org/XML</a>	XML
<a href="http://www.w3.org/CSS">www.w3.org/CSS</a>	Style sheets
<a href="http://www.w3.org/XSL">www.w3.org/XSL</a>	XML Style sheets

## §2. HTML

### I. Overview

- Hyper *Text* Markup Language
  - Designed for text
  - Origins of markup: publisher annotates manuscript with typesetting instructions (layout, typeface, boldness).
  - Tag-based

### II. Basic Structure of HTML Document

```

<!-- Basic web page -->
<html>
  <head>
    <title> Basic Web Page </title>
  </head>

  <body>
  </body>
</html>

```

html is the top-level element that contains the entire document. It has two subsections: the head and the body. The head contains information about the document. The body contains the content of the document.

There are two kinds of tags in HTML: *structure* tags and *presentation* tags.

#### 1. Structure tags

```

<head>          </head>
<title>         </title>
<body>         </body>

```

#### 2. Presentation tags

```

<center>       </ center >
<font>         </font>         (deprecated in HTML 4.0)
<b>            </b>

```

### III. Work Flow

#### 1. Editor

- Used to create the document.
- Pages can be created in a wide range of tools, from writing raw HTML in a basic text editor such as Notepad or vi, to a variety of WYSIWYG authoring tools that hide the HTML from you.

#### 2. Browser

- Interprets the tags and renders the document. (Presents you a view of the document.)
- Current browsers are very lenient. They accept sloppy HTML, and consequently are large, slow, and difficult to update

### IV. Terms

#### A. Element

- *name* and *content*
- Delimited by *start* and *end* tag
- Ex:
 

```
<title> Intro to SMIL </title>
```

The name of the element is `title`.  
 The content is the text "Intro to SMIL".

#### B. Attribute

- A parameter to an element.
  - An element can have 0, 1, or many attributes
- Specified in the start tag as *name* = "*value*"
- Enclose value in double quotes
- Ex:
 

```
<table border="0" >
```

The element name is `table`.  
 The name of the `table` attribute is `border`.  
 The value of `border` is `0`.

### C. Empty Element

- All the content information is specified in the attributes.
- Instead of an end tag, close with a forward slash:
- Ex:

```

```

The element name is `img`.

The name of the `img` attribute is `src`.

The value of `src` is `hockey.jpg`.

### D. Container Tags

The content lies between a start tag and an end tag.

### E. Parsing

Parsing is the act of scanning a document and interpreting the information based on the structure of the elements.

### F. Rendering

Rendering is the act of presenting a view of the information in a document. The presentation is in the form most appropriate to the environment. Browsers usually present a visual page, but they can also present a spoken or printed version. Use the `alt` attribute to specify an alternative rendering for purely visual items such as images.

## V. Style Sheets

- Separate structure from appearance.
- Set the rendering style for the HTML tags.
- Useful for maintaining consistent look across large websites.
- Specify style in the <head> of the HTML file.

### 1. Complete definition contained in the header

```
<head>
  <style type="text/css">
    body { color: maroon; background: white; }
    h1 { font-weight: bold; font-size: 150%;
        font-face:Arial; color:blue }
    dl { margin-left: 10%; margin-right: 10%; }
    dt { font-weight:bold; }
        table.contactInfo { color: maroon;
        background: rgb(204,204,255); }
  </style>
</head>
```

### 2. Complete definition contained in a separate file, and linked. Useful for modifying the appearance of an entire site by changing just one file.

```
<head>
  <link rel="stylesheet" href="styleEx4.css" />
  <title> HTML Example 4 </title>
</head>
```

File styleEx4.css contains:

```
<style type="text/css">
  body { color: black; }
  h1 { font-weight: bold; font-size: 150%;
      font-face: Arial; color: blue; }
  dl { margin-left: 10%; margin-right: 10%; }
  dt { font-weight:bold; }
      table.contactInfo { color: maroon;
      background: rgb(204,204,255); }
</style>
```

- Attribute **rel** specifies the relationship that the href attribute has to the document. Here we specify that href links to a style sheet.  
(For other possible relationships see <http://www.w3.org/TR/html4/types.html#type-links>)

## VI. Examples

### A. HTML/example1.html

```
<html>
  <head>
    <title> HTML Example 1 </title>
  </head>
  <body>
    <h1> <center>HTML Facts </center></h1>
    <table border="0">
      <tr> <td width="25%"> To: </td>
        <td> SMIL course attendees </td></tr>
      <tr> <td> From: </td> <td> Kathy B </td></tr>
      <tr> <td> Re: </td> <td> HTML </td></tr>
    </table>
    <dl>
      <dt> Editors </dt>
      <dd><ul>
        <li> Text editor </li>
        <li> WYSIWYG authoring tool </li>
      </ul></dd>
      <dt> Browsers </dt>
      <dd><ul>
        <li> Interpret the tags and render document </li>
        <li> Lenient about grammar </li>
      </ul></dd>
    </dl>
    <table>
      <tr> <td>Kathy </td></tr>
      <tr> <td>javakathy@teacher.com </td></tr>
      <tr> <td>Gotha, FL 34734 </td></tr>
    </table>
  </body>
</html>
```

**B. HTML/example2.html**

<!-- Example 2 shows sloppy HTML... missing end tags, case changes. Browser is able to compensate, and document renders just like Example 1. This puts a lot of burden on the browser. -->

```

<HTML>
  <head>
    <title> HTML Example 2: Sloppy HTML </title>
  </head>

  <bOdY>
    <h1> <center>HTML Facts </center></h1>
    <TABLE border="0">
      <tr> <td width="25%"> To:
        <td> SMIL course attendees
      <tr> <td> From: <td> Kathy B
      <tr> <td> Re: <td> HTML
    </table>
    <dl>
      <dt> Editors
      <dd><ul>
        <li> Text editor
        <li> WYSIWYG authoring tool
      </ul></dd>
      <p>
      <dt> Browsers
      <dd><ul>
        <li> Interpret the tags and render document
        <li> Lenient about grammar
      </ul></dd>
    </dl>

    <table>
      <tr> <td>Kathy
      <tr> <td>javakathy@teacher.com
      <tr> <td>Gotha, FL 34734
    </tAbLe>
  </BODY>
</html>

```

**C. HTML/example3.html**

```

<!-- Example 3 is the same text as Example 1.
Style information is specified in the header -->

<html >
  <head>
    <style type="text/css">
      body { color: maroon; background: white; }
      dl { margin-left: 10%; margin-right: 10%; }
      dt { font-weight:bold; }
      table.contactInfo { color: maroon;
        background: rgb(204,204,255); }
      h1 { font-weight: bold; font-size: 150%;
        font-face:Arial; color:blue }
    </style>
    <title> HTML Example 3: Style </title>
  </head>

  <body>
    <h1> <center>HTML Facts </center></h1>
    <table border="0">
      <tr> <td width="25%"> To: </td>
        <td> SMIL course attendees </td></tr>
      <tr> <td> From: </td> <td> Kathy B </td></tr>
      <tr> <td> Re: </td> <td> HTML </td></tr>
    </table>
    <dl>
      <dt> Editors </dt>
      <dd><ul>
        <li> Text editor </li>
        <li> WYSIWYG authoring tool </li>
      </ul></dd>
      <dt> Browsers </dt>
      <dd><ul>
        <li> Interpret the tags and render document </li>
        <li> Lenient about grammar </li>
      </ul></dd>
    </dl>

    <table class="contactInfo">
      <tr> <td>Kathy </td></tr>
      <tr> <td>javakathy@teacher.com </td></tr>
      <tr> <td>Gotha, FL 34734 </td></tr>
    </table>
  </body>
</html>

```

## VII. Summary

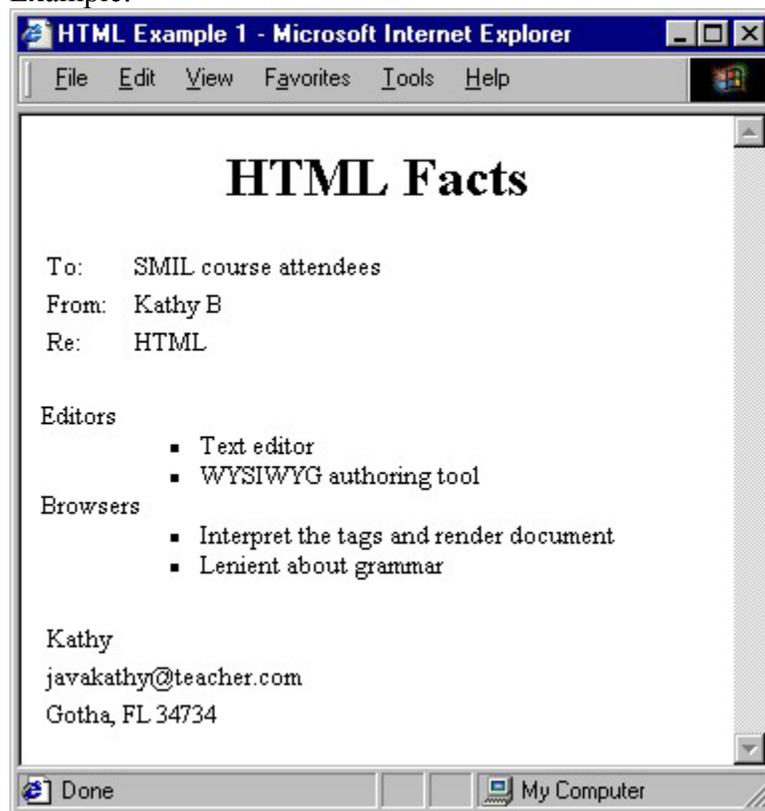
### A. Advantages

- non-proprietary format
- easy

### B. Disadvantages

- Structure and presentation intermixed  
Style sheets help, but don't completely eliminate this.
- No concern as to what kind of data the text is.

Example:



Memo signature contains name, email address, and some mailing address information. What if you wanted an automated way to extract that information into your address book? Can't if it's in HTML... it's just text formatted in a table... it has no meaning, even though we can look at it and see it is address information.

## §2. XML

### I. Overview

- Extensible Markup Language
  - No tags are predefined
  - Author creates the tags needed for the application
- Based on simplified SGML
  - Focus is on *structure* and *meaning* of the content

Example XML for Memo Signature:

```
<contactInfo>
  <name> Kathy B </name>
  <email> javakathy@teacher.com </email>
  <address>
    <city> Gotha </ city >
    <state> FL </state>
    <zip> 34734 </zip>
  </address>
</contactInfo>
```

- Questions:
  - If no tags are pre-defined, how do you determine what's a valid document?
  - And how do you render it???
- Answer:
  - The Document Type Definition

## II. DTD: Document Type Definition

- The “grammar” for your markup language.
- Describes every element that is allowed in a document, what the element attributes are, and the legal structure (which elements are allowed where).
- W3C provides DTDs for HTML, XHTML and SMIL
  - When using these markup languages, you don’t need to write your own DTDs. But if you can read them, you know exactly what is legal in a document.

### A. Specifying an Element

**<!ELEMENT *elementName* (*contentModel*) >**

The **<!ELEMENT** keyword begins an element declaration and the **>** character ends it. Between these are specified the element name and the content model.

- The element name will be the name used in the tag.
- The *content model* lists the children that are acceptable in that element. Children can be other elements, an enumerated type (list the possible values), or the symbol `PCDATA`, which stands for “Parsed Character Data”. Elements that have their content specified only via attributes are *empty elements*, and the content model is the keyword `EMPTY`.
- Occurrence indicators
  - + element must appear one or several times
  - \* element can appear 0 or more times
  - ? element can appear once or not at all
- connectors
  - , both elements to right and left of comma must appear, and in that order
  - | either left element or right element can appear (not both)

### B. Specifying an Attribute

**<!ATTLIST *elementName* *attributeName* *attributeType* *default* >**

The **<!ATTLIST** keyword begins an attribute declaration and the **>** character ends it. Between these are specified the element name, attribute name, attribute type, and an optional default value.

Example DTD for Memo Signature:

```
<!ELEMENT contactInfo (name,email*,address+)>

<!ELEMENT name (#PCDATA) >
<!ELEMENT email (#PCDATA) >
<!ATTLIST email location (home|office|other) "office">

<!ELEMENT address (street?,city,state,zip) >
<!ATTLIST address location (home|office|other) "office">
<!ELEMENT street (#PCDATA) >
<!ELEMENT city (#PCDATA) >
<!ELEMENT state (#PCDATA) >
<!ELEMENT zip (#PCDATA) >
```

We've defined a `contactInfo` that is composed of a name, any number of email addresses (optional), and at least one address. The name is a character string. The email is a character string that has a `location` attribute, which can be one of "home", "office", or "other".

An address consists of an optional street, a city, a state, and a zip. As for email, address that has a `location` attribute, which can be one of "home", "office", or "other". The elements `street`, `city`, `state`, and `zip` are character strings.

### III. Strict Syntax

To take some burden off of the browsers, XML is strict about the syntax.

- One top-level element that contains the rest
  - Required start and end tags
    - Empty element tags *must* be closed with a forward slash ("/")
  - Must put attributes in double quotes
  - Tags usually lowercase
  - Case-sensitive
- Can't do this in XML: `<BODY> yada, yada, yada </body>`

## IV. Terms

(From XML 1.0 Recommendation, <http://www.w3.org/XML>)

### A. Attribute

“An attribute is a parameter to an element declared in the DTD. An attribute's type and value range, including a possible default value, are defined in the DTD.”

### B. Element

“An element is a document structuring unit declared in the DTD. The element's content model is defined in the DTD, and additional semantics may be defined in the prose description of the element.”

### C. Validation

“Validation is a process whereby *documents* are verified against the associated DTD, ensuring that the structure, use of *elements*, and use of *attributes* are consistent with the definitions in the DTD.”

### D. Well-formed

“A *document* is well-formed when it is structured according to the rules defined in [Section 2.1](#) of the XML 1.0 Recommendation [[XML](#)]. Basically, this definition states that elements, delimited by their start and end tags, are nested properly within one another. “

## V. Namespace

If everyone is free to make up their own tags, chances are two applications will come up with the same name but slightly different meanings (attributes and rendering). Solution is to use a *namespace*, the equivalent of giving your tags a first and last name.

## VI. XHTML 1.0

- Latest version of HTML, 1/2000
- Reformulation of HTML4.01 (1997) in XM
- Three DTDs are available
  - `xhtml1-strict.dtd`
    - Must adhere to strict XML rules, including separating content from layout.
    - Use with CSS
  - `xhtml1-transitional.dtd`
    - For use when migrating pages that might still be accessed by older browsers.
  - `xhtml1-frameset.dtd`
    - Use when partitioning browser window into frames
- Encouraged to use XML declaration as first line of document

```
<?xml version="1.0" encoding="UTF-8"?>
```

- Place the appropriate DOCTYPE declaration in the document before the root element, depending on which DTD you are following:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"DTD/xhtml1-frameset.dtd">
```

### §3. Free Players

#### I. PC/Macintosh

GRiNS for SMIL 1.0, Version 1.5

<http://www.oratrix.com/GRiNS/gse/index.html>

QuickTime

<http://www.apple.com/quicktime>

RealPlayer 8 Basic

<http://www.real.com/player/index.html>

#### II. Unix

GRiNS for SMIL 1.0, Version 1.0

<http://www.oratrix.com/GRiNS/gsl/index.html>

Solaris

<http://www.oratrix.com/GRiNS/gsl/download/grins-sun-play-10.tgz>  
gunzip, then tar -xvf

RealPlayer Community Supported Player

<http://www.real.com/player/index.html>

#### Caveats from the Bleeding Edge

The challenge with any new technology is that the minute you put word to paper, it's out of date. The information written here is accurate as of early May 2001. The SMIL2.0 recommendation was released March 2001, but most players currently only support SMIL 1.0. Some commands that are known to be deprecated in SMIL 2.0 appear here because the players do not support the new commands yet.

Example:

<code>background-color="white"</code>	deprecated
<code>backgroundColor="white"</code>	SMIL 2.0; not implemented in most players yet.

## §4. SMIL

### I. Overview

#### A. What is SMIL good for?

1. Position content anywhere in your layout
2. Synchronize the timing of the elements
3. Display media to suit end-user's language, bit-rate, screen size, etc

#### B. Basic Structure of SMIL Document

```

<!-- DOCTYPE and DTD information -->
<smil>
  <head>
    <!-- information about presentation -->
    <meta name="Author" content="Kathy B" />
    <layout>
      <!-- main window information -->
      <!-- subregion height, width, & location -->
    </layout>
  </head>

  <body>
    <!-- multimedia presentation -->
  </body>
</smil>

```

`<smil>` is the top-level element that contains the entire document. Just as with HTML, it has 2 subsections: `<head>` and `<body>`. The `<head>` section contains information about the presentation. The `<body>` section contains the content of the presentation.

Comments are delimited by `<!--` and `-->` as in HTML and XML

The Document Type Declaration (DOCTYPE) goes before the top-level element. Both QuickTime and the GRiNS player are currently able to render a `.smil` document without this declaration. For SMIL 1.0, the DOCTYPE is

```

<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"
  "http://www.w3.org/TR/REC-smil/SMIL10.dtd">

```

## II. <head> Section

### A. <layout>

#### 1. Window Layout

##### a) root-layout

- Main window for your presentation
- Attributes
  1. **width, height** In pixels
  2. **background-color** (optional)
- Examples

```
< root-layout width="640" height="480" />
```

```
< root-layout width="640" height="480"
  background-color="black" />
```

##### b) region

- Sub-areas within main window where we'll place our media
- Origin (0,0) is at top left corner of main window.
- Attributes
  1. **width, height** In pixels, or as % of main window
  2. **top, left** In pixels, or as % of main window
  3. **id** Name by which we'll refer to region
  4. **background-color** (optional)
  5. **z-index** (optional)
  6. **fit** (optional)

##### (1) Exact position

Specify top left corner in pixels from main top left corner.

```
<region id="photo_region"
  width="640" height="480"
  top="20" left="80"
  background-color="black" />
```

##### (2) Relative position

Specify top left corner as % from main top left corner.

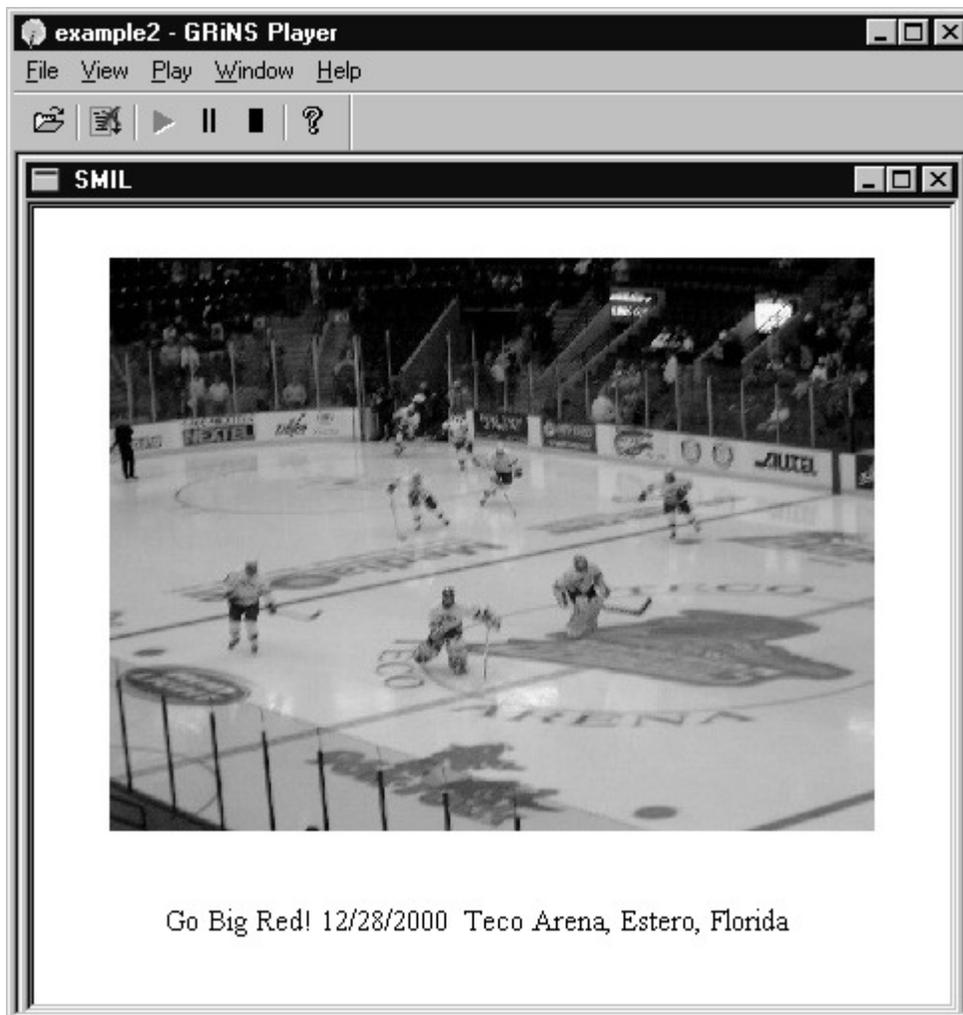
```
<region id="titlebar"
  width="200" height="100"
  top="25%" left="50%" />
```

Region starts a quarter of the way down from the top of main window, and halfway over from the left.

**ex2\_region.smil**

An image and a caption displayed in their own regions.

```
<smil>
  <head>
    <meta name="Author" content="Kathy Barshatzky" />
    <meta name="Copyright" content="javakathy.com" />
    <layout>
      <root-layout width="460" height="400" />
      <region id="photo_region"
        width="384" height="288" top="25" left="38" />
      <region id="caption_region"
        width="340" height="40" top="350" left="60" />
    </layout>
  </head>
  <body>
    <par>
      
      <text src="caption2.txt" alt="Teco Arena, Dec 2000"
        region="caption_region" dur="indefinite" />
    </par>
  </body>
</smil>
```



**(3) Overlapping regions**

You are allowed to overlap regions. The **z-index** attribute, which is optional and has a value of 0 by default, determines which region will be on top.

- Region with the greater z-index goes on top
- If two or more regions have the same z-index, the first region encountered is goes on the bottom, and subsequent regions are placed on top.

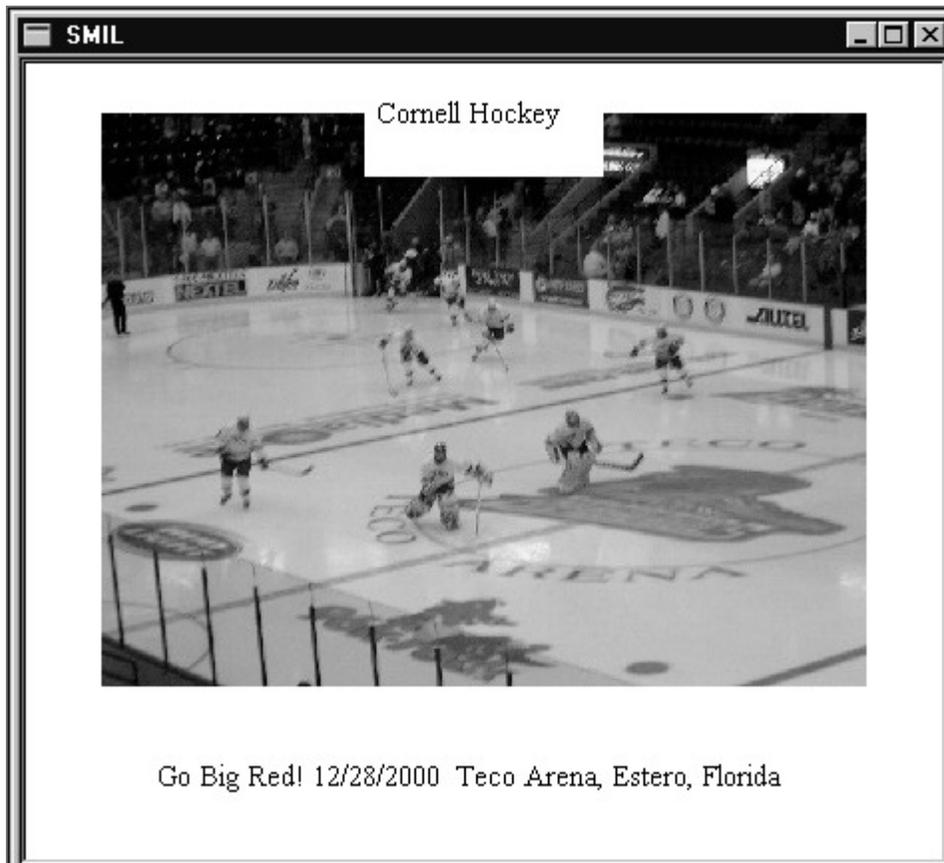
**ex3a\_overlap.smil**

```

<!-- In this example, the title region is given a z-index
greater than the default. Thus during rendering, it will
be placed over any overlapping regions of lower z-index.
-->
<smil>
  <head>
    <meta name="Author" content="Kathy Barshatzky" />
    <meta name="Copyright" content="javakathy.com" />
    <layout>
      <root-layout width="460" height="400" />
      <region id="photo_region"
        width="384" height="288" top="25" left="38" />
      <region id="caption_region"
        width="340" height="40" top="350" left="60" />
      <region id="title"
        width="120" height="40" top="17" left="170"
        background-color="white"
        z-index="10" />
    </layout>
  </head>
  <body>
    <par>
      <text src="title.txt" alt="Cornell Hockey"
        region="title" dur="indefinite"/>
      
      <text src="caption2.txt"
        region="caption_region" dur="indefinite"/>
    </par>
  </body>
</smil>

```

ex3a\_overlap.smil Screen Capture



The region with the title is placed over the region with the image.

**(4) fit attribute**

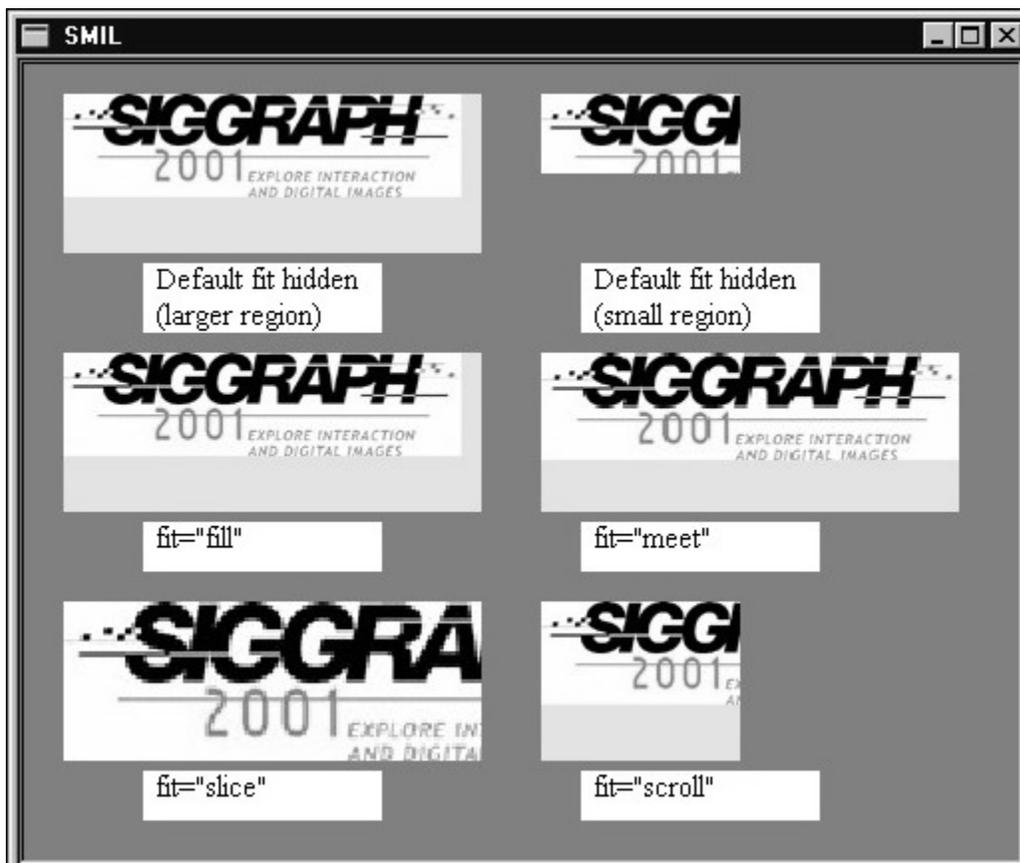
The `fit` attribute determines if and how the media, such as an image, is stretched to fill its region. The default value is “hidden”, which means that the image placed unstretched at the top left corner of the region. If the region is smaller than the image, the image is clipped.

- hidden** image placed unstretched at the top left corner of the region. Any portion of the image outside of the region is clipped.
- fill** stretches (disproportionately) the image to fit the region
- meet** stretches the image proportionately until it meets one of the image boundaries
- slice** stretches the image proportionally to fill the entire region. Any part of the image outside of the region is clipped.
- scroll** scrollbars appear if the image is larger than its region.

**fit\_example.smil** Screen Capture

Notice that the GRiNS Player v1.5 does not implement “fill” or “scroll”.

The regions have a yellow background so that you can tell their size and location.



**fit\_example.smil**

```

<!--
region's fit attribute
-->
<smil>
  <head>
    <meta name="Author" content="Kathy Barshatzky" />
    <meta name="Copyright" content="javakathy.com" />
  <layout>
    <root-layout width="500" height="400"
      background-color="gray"/>

    <region id="reg1"
      width="210" height="80" top="15" left="20"
      background-color="yellow" />
    <region id="reg1_txt" background-color="white"
      width="120" height="35" top="100" left="60" />

    <region id="reg2"
      width="100" height="40" top="15" left="260"
      background-color="yellow" />
    <region id="reg2_txt" background-color="white"
      width="120" height="35" top="100" left="280" />

    <region id="reg3" fit="fill"
      width="210" height="80" top="145" left="20"
      background-color="yellow" />
    <region id="reg3_txt" background-color="white"
      width="120" height="25" top="230" left="60" />

    <region id="reg4" fit="meet"
      width="210" height="80" top="145" left="260"
      background-color="yellow" />
    <region id="reg4_txt" background-color="white"
      width="120" height="25" top="230" left="280" />

    <region id="reg5" fit="slice"
      width="210" height="80" top="270" left="20"
      background-color="yellow" />
    <region id="reg5_txt" background-color="white"
      width="120" height="25" top="355" left="60" />

    <region id="reg6" fit="scroll"
      width="100" height="80" top="270" left="260"
      background-color="yellow" />
    <region id="reg6_txt" background-color="white"
      width="120" height="25" top="355" left="280" />

  </layout>
</head>

```

```
<body>
  <par>
    
    <text src="fit_default1.txt" alt="fit default"
         region="reg1_txt" dur="indefinite" />

    
    <text src="fit_default2.txt" alt="fit default"
         region="reg2_txt" dur="indefinite" />

    
    <text src="fit_fill.txt" alt="fit fill"
         region="reg3_txt" dur="indefinite" />

    
    <text src="fit_meet.txt" alt="fit meet"
         region="reg4_txt" dur="indefinite" />

    
    <text src="fit_slice.txt" alt="fit slice"
         region="reg5_txt" dur="indefinite" />

    
    <text src="fit_scroll.txt" alt="fit scroll"
         region="reg6_txt" dur="indefinite" />
  </par>
</body>
</smil>
```

## 2. Layout types

### a) default

```
<layout type="text/smil-basic-layout" >
```

<http://www.w3.org/TR/REC-smil/#layout-elements>

SMIL basic layout uses the formatting properties defined by CSS2 (Cascading Style Sheets Level 2) to control the layout of media object elements.

### b) CSS2

CSS2 can be used as an alternative to the basic layout.

```
<layout type="text/css">
  [region="r"] { top: 20px; left: 20px }
</layout>
```

<http://www.w3.org/TR/REC-smil/#layout>

## B. <meta>

- Attributes
  1. **name** is the *identifer* for the information specified by content
  2. **content** provides detailed information about the topic specified by name.

Examples:

```
<meta name="Author" content="Kathy Barshatzky" />
<meta name="Copyright" content="javakathy.com" />
```

### III. <body> Section

#### A. Media Object Elements

animation	time-based function of a target element
audio	audio clip, such as wav, mp3
img	still image, such as gif, jpg
ref	generic media reference
text	unformatted or html text
textstream	streaming text
video	video clip, such as Real movie, avi, mpg, QuickTime

Not all players support all possible media types. A complete list of MIME types is in the appendix for reference.

*continuous media* media objects with an intrinsic duration, such as video

*discrete media* media objects without an intrinsic duration, such as text

#### B. Text Media Types

##### 1. <text>

###### a) `type="text/plain"`

Plain, unformatted text in a .txt file.

###### b) `type="text/html"`

Example: Contents of the file caption3.html:

```
<html>
  <head>
    <title>Caption 3</title>
  </head>

  <body bgcolor="red" >
    <p>
      <b>Go Big Red!</b> 12/28/2000 Teco Arena, Estero, Florida
    </p>
  </body>
</html>
```

##### 2. <textstream>

textstream supports streaming text, such as RealText. See the References section for links to more information.

## C. Synchronization

- Two possible synchronizations:
  - `<par>` parallel      Media executed at the same time
  - `<seq>` sequential      Media executed one after the other

These can be nested. For example, you can have two sequences running in parallel:

```
<par>
  <seq>
  ...
</seq>

  <seq>
  ...
</seq>
</par>
```

- Attributes
  - dur**      duration      Length of time that the media is visible and playing.

```
<img dur="indefinite" ... />
<text dur="6s" ... />
```

Discrete media objects, such as text and images, should be given an appropriate duration, or they may vanish quickly from the window. The previous examples all had `dur = "indefinite"` so that the photo and caption would remain present.

- begin**      delay      Length of time that the media waits before playing.
 

```
<text begin="3s" ... />
```
- end**      delay      Time after which the media stops playing.
 

```
<text end="5s" ... />
```

**subtitles.smil**

This presentation plays a movie with subtitles. There are three elements playing in parallel: the video, the title, and the subtitles. The subtitles are a sequence of text files, each with the given duration.

```

<smil>
  <head>
    <meta name="Author" content="Kathy Barshatzky" />
    <meta name="Copyright" content="Visual Book Productions" />
    <layout>
      <root-layout width="400" height="300"
        background-color="black" />
      <region id="movie"
        width="360" height="202" left="20" top="12" />
      <region id="caption" background-color="white"
        width="300" height="40" left="36" top="260" />
      <region id="title" background-color="white"
        width="400" height="70" left="0" top="230" />
    </layout>
  </head>

  <body>
    <par>
      <video src="Clip35.avi" region="movie" alt="Clip 35" />
      <text src="title.txt" region="title" dur="indefinite" />
      <seq>
        <text region="caption" begin="3s"
          src="caption1.txt" dur="2s" />
        <text region="caption"
          src="data:,The Walls" dur="3s" />
        <text region="caption"
          src="caption2.txt" dur="4s" />
        <text region="caption"
          src="caption3.txt" dur="4s" />
      </seq>
    </par>
  </body>
</smil>

```

## D. Adapting the Presentation

### `<switch>`

Adapting a presentation to the end-user's system based on

```
system-bitrate
system-captions
system-language
system-overdub-or-caption
system-required
system-screen-size
system-screen-depth
```

A set of child test attributes are placed within the `<switch>` tags. The first match is executed. A match is the first child whose test attributes all evaluate to TRUE.

A complete description of the test attributes can be found at:

<http://www.w3.org/TR/REC-smil/#test>

For example, your presentation could play the audio track in different languages based on the user's preferred language:

```
...
<switch>
  <audio src="salesPitch-french" system-language="fr"/>
  <audio src=" salesPitch -english" system-language="en"/>
</switch>
```

If the user's preferred language is French, the French audio will play. If the user's preferred language is English, the English audio will play.

## E. Hyperlinking

A *link* has two ends, called *anchors*. The link starts at the source anchor and points to the destination anchor.

`<a href="xx">` sets up an anchor to a *complete* media object  
`<anchor>` sets up an anchor to a *portion of a* media object  
spatial or temporal subparts

Ex: Linking two text buttons to two other presentations:

```
<a href="presentation2.smil">  
  <text region="r_btn2" src="button2.txt" dur="indefinite" />  
</a>  
  
<a href="presentation3.smil">  
  <text region="r_btn3" src="button3.txt" dur="indefinite" />  
</a>
```

## §6. Reference Sites

### I. Recommendations/Specifications

<a href="http://www.w3.org/MarkUp">www.w3.org/MarkUp</a>	HTML
<a href="http://www.w3.org/AudioVideo">www.w3.org/AudioVideo</a>	SMIL
<a href="http://www.w3.org/XML">www.w3.org/XML</a>	XML
<a href="http://www.w3.org/CSS">www.w3.org/CSS</a>	Style sheets
<a href="http://www.w3.org/XSL">www.w3.org/XSL</a>	XML Style sheets

The following W3C recommendations are available for reference on the CD in the folder. Please check the sites for the latest revisions. Credit and copyrights belong to the original authors.

#### **Synchronized Multimedia Integration Language**

SMIL 2.0 W3C Working Draft 01 March 2001

References/smil10.doc	(Images in folder smil10_files)
References/SMIL10.dtd	
References/smil20.zip	
<a href="http://www.w3.org/TR/REC-smil/">http://www.w3.org/TR/REC-smil/</a>	SMIL 1.0
<a href="http://www.w3.org/TR/smil20/">http://www.w3.org/TR/smil20/</a>	SMIL 2.0, 1-Mar-2001

#### **XHTML™ 1.0: The Extensible HyperText Markup Language**

W3C Recommendation 26 January 2000

References/XHTML/xhtml11.pdf	
References/XHTML/xhtml11.zip	(contains the DTDs)
<a href="http://www.w3.org/TR/xhtml1/">http://www.w3.org/TR/xhtml1/</a>	
<a href="http://www.w3.org/TR/xhtml1/xhtml11.pdf">http://www.w3.org/TR/xhtml1/xhtml11.pdf</a>	

#### **Extensible Markup Language (XML) 1.0 (Second Edition)**

W3C Recommendation 6 October 2000

References/XML/REC-xml-20001006.pdf	
<a href="http://www.w3.org/XML">http://www.w3.org/XML</a>	

#### **Namespaces in XML**

W3C Consortium 14 January 1999

References/XML/NamespacesInXML.htm	
<a href="http://www.w3.org/TR/1999/REC-xml-names-19990114/">http://www.w3.org/TR/1999/REC-xml-names-19990114/</a>	

#### **MIME(Multipurpose Internet Mail Extensions) types**

References/MIMETypes/media-types.doc	
<a href="http://www.oac.uci.edu/indiv/ehood/MIME/">http://www.oac.uci.edu/indiv/ehood/MIME/</a>	

## II. Tutorials and More References

CWI (Centrum voor Wiskunde en Informatica)

Excellent SMIL presentation by Lloyd Rutledge and Lynda Hardman, members of the W3C working group that developed SMIL:

<http://www.cwi.nl/~media/SMIL/Tutorial/>

<http://www.cwi.nl/~media/publications/SMILTutorial.pdf>

RealSystem G2 Syntax Style

<http://service.real.com/help/library/blueprints/stylehtml/syntax.htm>

RealText Authoring Guide

This guide tells how to create and stream RealText. Last update: December 15, 2000

<http://docs.real.com/docs/smil/realtextauthoringguide8.pdf>

<http://service.real.com/help/library/encoders.html>

<http://service.real.com/help/library/guides/realtext/realtext.htm>

Cascading Style Sheets

<http://www.w3.org/Style/CSS/>

<http://www.w3.org/TR/REC-CSS2/>

<http://www.w3.org/TR/1998/REC-CSS2-19980512/css2.pdf>

<http://www.w3.org/MarkUp/Guide/Style.html>

Multipurpose Internet Mail Extensions (as in MIME content-type)

<http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1521.html>

Helio. Creators of Java-based SMIL Player, SOJA.

[www.helio.org/products/smil/tutorial/toc.html](http://www.helio.org/products/smil/tutorial/toc.html)

QuickTime-specific authoring

[www.apple.com/quicktime/authoring/qtsmil.html](http://www.apple.com/quicktime/authoring/qtsmil.html)

### III. Maillist

All administrative requests should be sent to the request address:

`www-smil-request@w3.org`

The `-request` mail address should be used for all list administrative requests. It accepts the following commands (in the Subject of an e-mail message):

`subscribe`           Subscribe to the list. If you want to subscribe under a different address, use a Reply-To: address header in the message.

`unsubscribe`        Unsubscribe from the list.

`help`                Get information about the mailing list.

`archive help`        Get information about the list archive(s).

### IV. Authoring Tools

#### A. GRiNS

<http://www.oratrix.com/GRiNS/index.html>

#### B. Real

<http://www.real.com/>

## **Reference Material**

§Appendix-1. SMIL 1.0 Specification

§Appendix-2. MIME Registered Media Types



REC-smil-19980615

# Synchronized Multimedia Integration Language (SMIL) 1.0 Specification

W3C Recommendation 15-June-1998

This version:

<http://www.w3.org/TR/1998/REC-smil-19980615>

Latest version:

<http://www.w3.org/TR/REC-smil>

Previous version:

<http://www.w3.org/TR/1998/PR-smil-19980409>

Copyright © 1998 W3C (MIT, INRIA, Keio), All Rights Reserved. W3C [liability](#), [trademark](#), [document use](#) and [software licensing](#) rules apply. Your interactions with this site are in accordance with our [public](#) and [Member](#) privacy statements.

## About this Document

This document has been prepared by the Synchronized Multimedia Working Group (WG) of the World Wide Web Consortium. The WG included the following individuals:

- Stephan Bugaj, Lucent/Bell Labs
- Dick Bulterman, CWI
- Bruce Butterfield, RealNetworks
- Wo Chang, NIST
- Guy Fouquet, Alcatel
- Christian Gran, GMD
- Mark Hakkinen, The Productivity Works
- Lynda Hardman, CWI
- Peter Hoddie, Apple
- Klaus Hofrichter, GMD
- Philipp Hoschka, W3C
- Jack Jansen, CWI
- George Kerscher, DAISY Consortium
- Rob Lanphier, RealNetworks
- Nabil Layaïda, INRIA
- Stephanie Leif, RealNetworks
- Sjoerd Mullender, CWI
- Didier Pillet, CNET/DSM
- Anup Rao, Netscape
- Lloyd Rutledge, CWI
- Patrick Soquet, Havas
- Warner ten Kate, Philips
- Jacco van Ossenbruggen, CWI
- Michael Vernick, Lucent/Bell Labs

- Jin Yu, DEC

**Acknowledgments:** In addition to the working group members, the following people contributed to the SMIL effort: Bert Bos (W3C), Dan Connolly (W3C), Patrick Deunhouwer (Philips), Martin Dürst (W3C), Al Gilman, Håkon Lie (W3C), Chris Lilley (W3C), Curtis Reynolds (RealNetworks), Michael Riesman, Curtis Reynolds (RealNetworks), Henning Schulzrinne (Columbia University) and Koga Youichirou (W3C).

**Editor:** Philipp Hoschka, W3C ([hoschka@w3.org](mailto:hoschka@w3.org))

## Abstract

This document specifies version 1 of the Synchronized Multimedia Integration Language (SMIL 1.0, pronounced "smile"). SMIL allows integrating a set of independent multimedia objects into a synchronized multimedia presentation. Using SMIL, an author can

1. describe the temporal behavior of the presentation
2. describe the layout of the presentation on a screen
3. associate hyperlinks with media objects

This specification is structured as follows: Section 1 presents the specification approach. Section 2 defines the "smil" element. Section 3 defines the elements that can be contained in the head part of a SMIL document. Section 4 defines the elements that can be contained in the body part of a SMIL document. In particular, this Section defines the time model used in SMIL. Section 5 describes the SMIL DTD.

## Status of this Document

This document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

Comments on this Recommendation may be sent to the [public mailing list www-smil@w3.org](mailto:public mailing list www-smil@w3.org).

## Available languages

The English version of this specification is the only normative version. However, for translations in other languages see <http://www.w3.org/AudioVideo/SMIL/translations>.

## Errata

The list of known errors in this specification is available at <http://www.w3.org/AudioVideo/SMIL/errata>.

## Table of Contents

- [1 Specification Approach](#)
- [2 The smil Element](#)
- [3 The Document Head](#)
  - [3.1 The head Element](#)
  - [3.2 The layout Element](#)
  - [3.3 SMIL Basic Layout Language](#)
    - [3.3.1 The region Element](#)
    - [3.3.2 The root-layout Element](#)
  - [3.4 The meta Element](#)
- [4 The Document Body](#)
  - [4.1 The body Element](#)
  - [4.2 Synchronization Elements](#)
    - [4.2.1 The par Element](#)
    - [4.2.2 The seq Element](#)
    - [4.2.3 Media Object Element](#): The ref, animation, audio, img, video, text and textstream elements
    - [4.2.4 SMIL Time Model](#)
      - [4.2.4.1 Time Model Values](#)
      - [4.2.4.2 Determining Values of Model Values for SMIL 1.0 Elements](#)
  - [4.3 The switch Element](#)
  - [4.4 Test Attributes](#)
  - [4.5 Hyperlinking](#) Elements
    - [4.5.1 The a Element](#)
    - [4.5.2 The anchor Element](#)
- [5 SMIL DTD](#)
  - [5.1 Relation to XML](#)
  - [5.2 DTD](#)
- [Appendix](#)
  - [Extending SMIL 1.0](#)
  - [Using SMIL 1.0 as an Extension](#)

## 1 Specification Approach

SMIL documents are XML 1.0 documents [[XML10](#)]. The reader is expected to be familiar with the concepts and terms defined in XML 1.0.

This specification does not rely on particular features defined in URLs that cannot potentially be expressed using URNs. Therefore, the more generic term URI [[URI](#)] is used throughout the specification.

The syntax of SMIL documents is defined by the DTD in [Section 5.2](#). The syntax of an attribute value that cannot be defined using the DTD notation is defined together with the first element using an attribute that can contain the attribute value. The syntax of such attribute values is defined using the Extended Backus-Naur Form (EBNF) defined in the XML 1.0 specification.

An element definition is structured as follows: First, all attributes of the element are defined in alphabetical order. An attribute is defined in the following way: If the attribute is used by an element for the first time in the specification, the semantics of the attribute are defined. If the attribute has already been used by another element, the specification refers to the definition of the attribute in the first element that used it. The definition of element attributes is followed by the definition of any attribute values whose syntax cannot be defined using the DTD notation. The final section in an element definition specifies the element content.

## 2 The `smil` Element

### Element Attributes

The "smil" element can have the following attribute:

id

This attribute uniquely identifies an element within a document. Its value is an XML identifier.

### Element Content

The "smil" element can contain the following children:

body

Defined in [Section 4.1](#)

head

Defined in [Section 3.1](#)

## 3 The Document Head

### 3.1 The `head` Element

The "head" element contains information that is not related to the temporal behavior of the presentation.

#### Element Attributes

The "head" element can have the following attribute:

id

Defined in [Section 2](#)

#### Element Content

The "head" element can contain the following children:

layout

Defined in [Section 3.2](#)

meta

Defined in [Section 3.4](#)

switch

Defined in [Section 4.3](#)

The "head" element may contain any number of "meta" elements and either a "layout" element or a "switch" element.

### 3.2 The `layout` Element

The "layout" element determines how the elements in the document's body are positioned on an abstract rendering surface (either visual or acoustic).

If a document contains no layout element, the positioning of the body elements is implementation-dependent.

A SMIL document can contain multiple alternative layouts by enclosing several layout elements within a "switch" element (defined in [Section 4.3](#)). This can be used for example to describe the document's layout using different layout languages.

The following example shows how CSS2 can be used as alternative to the SMIL basic layout language (defined in [Section 3.3](#)):

```
<smil>
  <head>
    <switch>
      <layout type="text/css">
        [region="r"] { top: 20px; left: 20px }
      </layout>
      <layout>
        <region id="r" top="20" left="20" />
      </layout>
    </switch>

  </head>

  <body>
    <seq>
      
    </seq>
  </body>
</smil>
```

(note that in this example, both layout alternatives result in the same layout)

### Element Attributes

id

Defined in [Section 2](#)

type

This attribute specifies which layout language is used in the layout element. If the player does not understand this language, it must skip all content up until the next "</layout>" tag. The default value of the type attribute is "text/smil-basic-layout".

### Element Content

If the type attribute of the layout element has the value "text/smil-basic-layout", it can contain the following elements:

region

Defined in [Section 3.3.1](#)

root-layout

Defined in [Section 3.3.2](#)

If the type attribute of the "layout" element has another value, the element contains character data.

### 3.3 SMIL Basic Layout Language

This section defines a basic layout language for SMIL. SMIL basic layout is consistent with the visual rendering model defined in CSS2, it reuses the formatting properties defined by the CSS2 specification, and newly introduces the `fit` attribute [CSS2]. The reader is expected to be familiar with the concepts and terms defined in CSS2.

SMIL basic layout only controls the layout of media object elements (defined in [Section 4.2.3](#)). It is illegal to use SMIL basic layout for other SMIL elements.

The type identifier for SMIL basic layout is `text/smil-basic-layout`.

#### Fixed Property Values

The following stylesheet defines the values of the CSS2 properties "display" and "position" that are valid in SMIL basic layout. These property values are fixed:

```

a          {display:block}
anchor     {display:block}
animation  {display: block;
            position: absolute}
body       {display: block}
head       {display: none}
img        {display: block;
            position: absolute}
layout     {display: none}
meta       {display: none}
par        {display: block}
region     {display: none}
ref        {display: block;
            position: absolute}
root-layout {display: none}
seq        {display: block}
smil       {display: block}
switch     {display:block}
text       {display: block;
            position: absolute}
textstream {display: block;
            position: absolute}
video      {display: block;
            position: absolute}

```

Note that as a result of these definitions, all absolutely positioned elements (animation, img, ref, text, textstream and video) are contained within a single containing block defined by the content content edge of the root element (smil).

#### Default Values

SMIL basic layout defines default values for all layout-related attributes. These are consistent with the initial values of the corresponding properties in CSS2.

If the author wants to select the default layout values for *all* media object elements in a document, the document must contain an empty layout element of type "text/smil-basic-layout" such as:

```
<layout type="text/smil-basic-layout"></layout>
```

### 3.3.1 The `region` Element

The `region` element controls the position, size and scaling of media object elements.

In the following example fragment, the position of a text element is set to a 5 pixel distance from the top border of the rendering window:

```
<smil>
  <head>
    <layout>
      <region id="a" top="5" />
    </layout>
  </head>
  <body>
    <text region="a" src="text.html" dur="10s" />
  </body>
</smil>
```

#### Element Attributes

The "region" element can have the following attributes:

##### `background-color`

The use and definition of this attribute are identical to the `background-color` property in the CSS2 specification, except that SMIL basic layout does not require support for "system colors". If the `background-color` attribute is absent, the background is transparent.

##### `fit`

This attribute specifies the behavior if the intrinsic height and width of a visual media object differ from the values specified by the height and width attributes in the "region" element. This attribute does not have a 1-1 mapping onto a CSS2 property, but can be simulated in CSS2. This attribute can have the following values:

##### `fill`

Scale the object's height and width independently so that the content just touches all edges of the box.

##### `hidden`

- If the intrinsic height (width) of the media object element is smaller than the height (width) defined in the "region" element, render the object starting from the top (left) edge and fill up the remaining height (width) with the background color.
- If the intrinsic height (width) of the media object element is greater than the height (width) defined in the "region" element, render the object starting from the top (left) edge until the height (width) defined in the "region" element is reached, and clip the parts of the object below (right of) the height (width).

##### `meet`

Scale the visual media object while preserving its aspect ratio until its height or width is equal to the value specified by the height or width attributes, while none of the content is clipped. The object's left top corner is positioned at the top-left coordinates of the box, and empty space at the left or bottom is filled up with the background color.

##### `scroll`

A scrolling mechanism should be invoked when the element's rendered contents exceed its bounds.

**slice**

Scale the visual media object while preserving its aspect ratio so that its height or width are equal to the value specified by the height and width attributes while some of the content may get clipped. Depending on the exact situation, either a horizontal or a vertical slice of the visual media object is displayed. Overflow width is clipped from the right of the media object. Overflow height is clipped from the bottom of the media object.

The default value of `fill` is hidden.

**height**

The use and definition of this attribute are identical to the "height" property in the CSS2 specification. Attribute values can be "percentage" values, and a variation of the "length" values defined in CSS2. For "length" values, SMIL basic layout only supports pixel units as defined in CSS2. It allows to leave out the "px" unit qualifier in pixel values (the "px" qualifier is required in CSS2).

**id**

Defined in [Section 2](#)

A region element is applied to a positionable element by setting the [region](#) attribute of the positionable element to the id value of the region.

The "id" attribute is required for "region" elements.

**left**

The use and definition of this attribute are identical to the "left" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute.

The default value is zero.

**skip-content**

This attribute is introduced for future extensibility of SMIL (see [Appendix](#)). It is interpreted in the following two cases:

- If a new element is introduced in a future version of SMIL, and this element allows SMIL 1.0 elements as element content, the "skip-content" attribute controls whether this content is processed by a SMIL 1.0 player.
- If an empty element in SMIL version 1.0 becomes non-empty in a future SMIL version, the "skip-content" attribute controls whether this content is ignored by a SMIL 1.0 player, or results in a syntax error.

If the value of the "skip-content" attribute is "true", and one of the cases above apply, the content of the element is ignored. If the value is "false", the content of the element is processed.

The default value for "skip-content" is "true".

**title**

This attribute offers advisory information about the element for which it is set. Values of the title attribute may be rendered by user agents in a variety of ways. For instance, visual browsers frequently display the title as a "tool tip" (a short message that appears when the pointing device pauses over an object).

It is strongly recommended that all "region" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

**top**

The use and definition of this attribute are identical to the "top" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute.

The default value is zero.

**width**

The use and definition of this attribute are identical to the "width" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute.

**z-index**

The use and definition of this attribute are identical to the "z-index" property in the CSS2 specification, with the following exception:

- If two boxes generated by elements A and B have the same stack level, then
  1. If the display of an element A starts later than the display of an element B, the box of A is stacked on top of the box of B (temporal order).
  2. If the display of the elements starts at the same time, and an element A occurs later in the SMIL document text than an element B, the box of A is stacked on top of the box of B (document tree order as defined in CSS2).

**Element Content**

"region" is an empty element.

**3.3.2 The root-layout element**

The `root-layout` element determines the value of the layout properties of the root element, which in turn determines the size of the viewport, e.g. the window in which the SMIL presentation is rendered.

If a document contains more than one `root-layout` element, this is an error, and the document should not be displayed.

**Element Attributes**

The `root-layout` element can have the following attributes:

**background-color**

Defined in [Section 3.3.1](#)

**height**

Defined in [Section 3.3.1](#)

Sets the height of the root element. Only length values are allowed.

**id**

Defined in [Section 2](#)

**skip-content**

Defined in [Section 3.3.1](#)

**title**

Defined in [Section 3.3.1](#)

**width**

Defined in [Section 3.3.1](#)

Sets the width of the root element. Only length values are allowed.

**Element Content**

`root-layout` is an empty element.

### 3.4 The `meta` Element

The `meta` element can be used to define properties of a document (e.g., author, expiration date, a list of key words, etc.) and assign values to those properties. Each `meta` element specifies a single property/value pair.

#### Element Attributes

The `meta` element can have the following attributes:

`content`

This attribute specifies the value of the property defined in the `meta` element.  
The "content" attribute is required for "meta" elements.

`id`

Defined in [Section 2](#)

`name`

This attribute identifies the property defined in the `meta` element.  
The "name" attribute is required for "meta" elements.

`skip-content`

Defined in [Section 3.3.1](#)

The list of properties is open-ended. This specification defines the following properties:

`base`

The value of this property determines the base URI for all relative URIs used in the document.

`pics-label` or PICS-Label

The value of this property specifies a valid rating label for the document as defined by PICS [\[PICS\]](#).

`title`

The value of this property contains the title of the presentation.

#### Element Content

`meta` is an empty element.

## 4 The Document Body

### 4.1 The `body` Element

The `body` element contains information that is related to the temporal and linking behavior of the document. It implicitly defines a `seq` element (defined in Section 4.2.2, see Section 4.2.4 for a definition of the temporal semantics of the `body` element).

#### Element Attributes

The `body` element can have the following attribute:

`id`  
Defined in [Section 2](#)

#### Element Content

The `body` element can contain the following children:

`a`  
Defined in [Section 4.5.1](#)

`animation`  
Defined in [Section 4.2.3](#)

`audio`  
Defined in [Section 4.2.3](#)

`img`  
Defined in [Section 4.2.3](#)

`par`  
Defined in [Section 4.2.1](#)

`ref`  
Defined in [Section 4.2.3](#)

`seq`  
Defined in [Section 4.2.2](#)

`switch`  
Defined in [Section 4.3](#)

`text`  
Defined in [Section 4.2.3](#)

`textstream`  
Defined in [Section 4.2.3](#)

`video`  
Defined in [Section 4.2.3](#)

## 4.2 Synchronization Elements

### 4.2.1 The `par` Element

The children of a `par` element can overlap in time. The textual order of appearance of children in a `par` has no significance for the timing of their presentation.

#### Element Attributes

The `par` element can have the following attributes:

##### `abstract`

A brief description of the content contained in the element.

##### `author`

The name of the author of the content contained in the element.

##### `begin`

This attribute specifies the time for the explicit begin of an element. See [Section 4.2.4](#) for a definition of its semantics.

The attribute can contain the following two types of values:

##### `delay-value`

A delay value is a clock-value measuring presentation time. Presentation time advances at the speed of the presentation. It behaves like the timecode shown on a counter of a tape-deck. It can be stopped, decreased or increased either by user actions, or by the player itself.

The semantics of a delay value depend on the element's first ancestor that is a synchronization element (i.e. ancestors that are "a" or "switch" elements are ignored):

- If this ancestor is a "par" element, the value defines a delay from the effective begin of that element (see Figure 4.1).
- If this ancestor is a "seq" element (defined in [Section 4.2.2](#)), the value defines a delay from the effective end of the first lexical predecessor that is a synchronization element (see Figure 4.2).

##### `event-value`

The element begins when a certain event occurs (see Figure 4.3). Its value is an element-event (see Definition below).

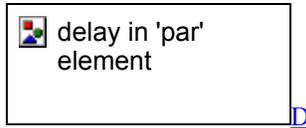
The element generating the event must be "in scope". The set of "in scope" elements S is determined as follows:

1. Take all children from the element's first ancestor that is a synchronization element and add them to S.
2. Remove all "a" and "switch" elements from S. Add the children of all "a" elements to S, unless they are "switch" elements.

The resulting set S is the set of "in scope" elements.

---

```
<par>
  <audio id="a" begin="6s" src="audio" />
</par>
```

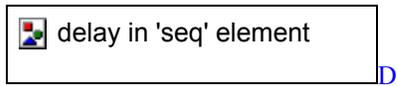



---

*Figure 4.1: Using a delay value within a "par" element*

---

```
<seq>
  <audio src="audio1" />
  <audio begin="5s" src="audio2" />
</seq>
```

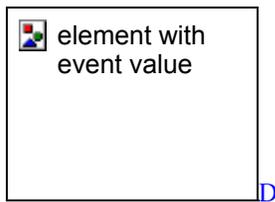



---

*Figure 4.2: Using a delay value within a "seq" element*

---

```
<par>
  <audio id="a" begin="6s" ... />
  <img begin="id(a)(4s)" ... />
</par>
```




---

*Figure 4.3: Synchronization attribute with element event value*

---

copyright

The copyright notice of the content contained in the element.

dur

This attribute specifies the explicit duration of an element. See [Section 4.2.4](#) for a definition of its semantics. The attribute value can be a clock value, or the string "indefinite".

end

This attribute specifies the explicit end of an element. See [Section 4.2.4](#) for a definition of its semantics. The attribute can contain the same types of attribute values as the "begin" attribute.

endsync

For a definition of the semantics of this attribute, see [Section 4.2.4](#). The attribute can have the following values:

- **first**  
For a definition of the semantics of this value, see [Section 4.2.4](#).

- `id-ref`  
This attribute value has the following syntax:  
  
`id-ref ::= "id(" id-value ")"`  
where "id-value" must be a legal XML identifier.  
For a definition of the semantics of this value, see [Section 4.2.4](#).
- `last`  
For a definition of the semantics of this value, see [Section 4.2.4](#).

The default value of `endsync` is `last`.

`id`

Defined in [Section 2](#)

`region`

This attribute specifies an abstract rendering surface (either visual or acoustic) defined within the layout section of the document. Its value must be an XML identifier. If no rendering surface with this `id` is defined in the layout section, the values of the formatting properties of this element are determined by the default layout.

The "region" attribute on "par" elements cannot be used by the basic layout language for SMIL defined in this specification. It is added for completeness, since it may be required by other layout languages.

`repeat`

For a definition of the semantics of this attribute, see [Section 4.2.4](#). The attribute value can be an integer, or the string "indefinite". The default value is 1.

`system-bitrate` Defined in [Section 4.4](#)

`system-captions` Defined in [Section 4.4](#)

`system-language` Defined in [Section 4.4](#)

`system-overdub-or-caption` Defined in [Section 4.4](#)

`system-required` Defined in [Section 4.4](#)

`system-screen-size` Defined in [Section 4.4](#)

`system-screen-depth` Defined in [Section 4.4](#)

`title` Defined in [Section 3.3.1](#)

It is strongly recommended that all `par` elements have a `title` attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

### Note on Synchronization between Children

The accuracy of synchronization between the children in a parallel group is implementation-dependent. Take the example of synchronization in case of playback delays, i.e. the behavior when the `par` element contains two or more continuous media types such as audio or video, and one of them experiences a delay. A player can show the following synchronization behaviors:

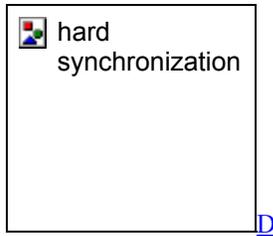
**hard synchronization**

The player synchronizes the children in the `par` element to a common clock (see Figure 4.4 a)).

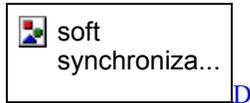
**soft synchronization**

Each child of the `par` element has its own clock, which runs independently of the clocks of other children in the `par` element (see Figure 4.4 b)).

---



a) *hard synchronization: Delay in video: Either the audio is stopped, or some video frames are dropped. The exact behavior is implementation-dependent*



b) *soft synchronization*

---

Figure 4.4: Effect of a delay on playout schedule for players using different synchronization policies

### Attribute Values

#### clock value

Clock values have the following syntax:

```

Clock-val          ::= Full-clock-val | Partial-clock-val |
Timecount-val
Full-clock-val     ::= Hours ":" Minutes ":" Seconds ( "."
Fraction)?
Partial-clock-val  ::= Minutes ":" Seconds ( "." Fraction)?
Timecount-val     ::= Timecount ( "." Fraction)?
                   ("h" | "min" | "s" | "ms")? ; default
is "s"
Hours              ::= 2DIGIT; any positive number
Minutes           ::= 2DIGIT; range from 00 to 59
Seconds           ::= 2DIGIT; range from 00 to 59
Fraction          ::= DIGIT+
Timecount         ::= DIGIT+
2DIGIT            ::= DIGIT DIGIT
DIGIT             ::= [0-9]

```

The following are examples of legal clock values:

- Full clock value: 02:30:03 = 2 hours, 30 minutes and 3 seconds
- Partial clock value: 02:33 = 2 minutes and 33 seconds
- Timecount values:
  - 3h = 3 hours
  - 45min = 45 minutes
  - 30s = 30 seconds
  - 5ms = 5 milliseconds

A fraction x with n digits represents the following value:  $x * 1/10^{**n}$

Examples:

00.5s = 5 \* 1/10 seconds = 500 milliseconds  
 00:00.005 = 5 \* 1/1000 seconds = 5 milliseconds

element-event value

An *element event* value specifies a particular event in a synchronization element.

An element event has the following syntax:

```
Element-event ::= "id(" Event-source ")(" Event ")"
Event-source  ::= Id-value
Event         ::= "begin" | Clock-val | "end"
```

The following events are defined:

**begin**

This event is generated at an element's effective begin.

Example use: `begin="id(x)(begin)"`

**clock-val**

This event is generated when a clock associated with an element reaches a particular value. This clock starts at 0 at the element's effective begin. For `par` and `seq` elements, the clock gives the presentation time elapsed since the effective begin of the element. For media object elements, the semantics are implementation-dependent. The clock may either give presentation time elapsed since the effective begin, or it may give the media time of the object. The latter may differ from the presentation time that elapsed since the object's display was started e.g. due to rendering or network delays, and is the recommended approach.

It is an error to use a clock value that exceeds the value of the effective duration of the element generating the event.

Example use: `begin="id(x)(45s)"`

**end**

This event is generated at the element's effective end.

Example use: `begin="id(x)(end)"`

## Element Content

The `par` element can contain the following children:

<code>a</code>	Defined in <a href="#">Section 4.5.1</a>
<code>animation</code>	Defined in <a href="#">Section 4.2.3</a>
<code>audio</code>	Defined in <a href="#">Section 4.2.3</a>
<code>img</code>	Defined in <a href="#">Section 4.2.3</a>
<code>par</code>	Defined in <a href="#">Section 4.2.1</a>
<code>ref</code>	Defined in <a href="#">Section 4.2.3</a>
<code>seq</code>	Defined in <a href="#">Section 4.2.2</a>
<code>switch</code>	Defined in <a href="#">Section 4.3</a>
<code>text</code>	Defined in <a href="#">Section 4.2.3</a>
<code>textstream</code>	Defined in <a href="#">Section 4.2.3</a>
<code>video</code>	Defined in <a href="#">Section 4.2.3</a>

All of these elements may appear multiple times as direct children of a `par` element.

### 4.2.2 The **seq** Element

The children of a `seq` element form a temporal sequence.

#### Attributes

The `seq` element can have the following attributes:

<code>abstract</code>	Defined in <a href="#">Section 4.2.1</a>
<code>author</code>	Defined in <a href="#">Section 4.2.1</a>
<code>begin</code>	Defined in <a href="#">Section 4.2.1</a>
<code>copyright</code>	Defined in <a href="#">Section 4.2.1</a>
<code>dur</code>	Defined in <a href="#">Section 4.2.1</a>
<code>end</code>	Defined in <a href="#">Section 4.2.1</a>
<code>id</code>	Defined in <a href="#">Section 2</a>
<code>region</code>	Defined in <a href="#">Section 4.2.1</a>

The `region` attribute on `seq` elements cannot be used by the basic layout language for SMIL defined in this specification. It is added for completeness, since it may be required by other layout languages.

<code>repeat</code>	Defined in <a href="#">Section 4.2.1</a>
<code>system-bitrate</code>	Defined in <a href="#">Section 4.4</a>
<code>system-captions</code>	Defined in <a href="#">Section 4.4</a>
<code>system-language</code>	Defined in <a href="#">Section 4.4</a>
<code>system-overdub-or-caption</code>	Defined in <a href="#">Section 4.4</a>
<code>system-required</code>	Defined in <a href="#">Section 4.4</a>
<code>system-screen-size</code>	Defined in <a href="#">Section 4.4</a>
<code>system-screen-depth</code>	Defined in <a href="#">Section 4.4</a>
<code>title</code>	Defined in <a href="#">Section 3.3.1</a>

It is strongly recommended that all `seq` elements have a `title` attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

#### Element Content

The `seq` element can contain the following children:

<code>a</code>	Defined in <a href="#">Section 4.5.1</a>
<code>animation</code>	Defined in <a href="#">Section 4.2.3</a>
<code>audio</code>	Defined in <a href="#">Section 4.2.3</a>
<code>img</code>	Defined in <a href="#">Section 4.2.3</a>
<code>par</code>	Defined in <a href="#">Section 4.2.1</a>
<code>ref</code>	Defined in <a href="#">Section 4.2.3</a>
<code>seq</code>	Defined in <a href="#">Section 4.2.2</a>
<code>switch</code>	Defined in <a href="#">Section 4.3</a>
<code>text</code>	Defined in <a href="#">Section 4.2.3</a>
<code>textstream</code>	Defined in <a href="#">Section 4.2.3</a>
<code>video</code>	Defined in <a href="#">Section 4.2.3</a>

### 4.2.3 Media Object Elements: The `ref`, `animation`, `audio`, `img`, `video`, `text` and `textstream` elements

The media object elements allow the inclusion of media objects into a SMIL presentation. Media objects are included by reference (using a URI).

There are two types of media objects: media objects with an intrinsic duration (e.g. video, audio) (also called "continuous media"), and media objects without intrinsic duration (e.g. text, image) (also called "discrete media").

Anchors and links can be attached to visual media objects, i.e. media objects rendered on a visual abstract rendering surface.

When playing back a media object, the player must not derive the exact type of the media object from the name of the media object element. Instead, it must rely solely on other sources about the type, such as type information contained in the "type" attribute, or the type information communicated by a server or the operating system.

Authors, however, should make sure that the group into which of the media object falls (animation, audio, img, video, text or textstream) is reflected in the element name. This is in order to increase the readability of the SMIL document. When in doubt about the group of a media object, authors should use the generic "ref" element.

#### Element Attributes

Media object elements can have the following attributes:

`abstract`

Defined in [Section 4.2.1](#)

`alt`

For user agents that cannot display a particular media-object, this attribute specifies alternate text. It is strongly recommended that all media object elements have an "alt" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

`author`

Defined in [Section 4.2.1](#)

`begin`

Defined in [Section 4.2.1](#)

`clip-begin`

The clip-begin attribute specifies the beginning of a sub-clip of a continuous media object as offset from the start of the media object.

Values in the clip-begin attribute have the following syntax:

```
Clip-time-value ::= Metric "=" ( Clock-val | Smpte-val )
Metric          ::= Smpte-type | "npt"
Smpte-type     ::= "smpte" | "smpte-30-drop" | "smpte-25"
Smpte-val      ::= Hours ":" Minutes ":" Seconds
                [ ":" Frames [ "." Subframes ] ]

Hours          ::= 2DIGIT
Minutes        ::= 2DIGIT
Seconds        ::= 2DIGIT
Frames         ::= 2DIGIT
Subframes      ::= 2DIGIT
```

The value of this attribute consists of a metric specifier, followed by a time value whose syntax and semantics depend on the metric specifier. The following formats are allowed:

#### SMPTE Timestamp

SMPTE time codes [\[SMPTE\]](#) can be used for frame-level access accuracy. The metric specifier can have the following values:

smpte

smpte-30-drop

These values indicate the use of the "SMPTE 30 drop" format with 29.97 frames per second. The "frames" field in the time value can assume the values 0 through 29. The difference between 30 and 29.97 frames per second is handled by dropping the first two frame indices (values 00 and 01) of every minute, except every tenth minute.

smpte-25

The "frames" field in the time specification can assume the values 0 through 24.

The time value has the format hours:minutes:seconds:frames.subframes. If the frame value is zero, it may be omitted. Subframes are measured in one-hundredth of a frame.

Examples:

```
clip-begin="smpte=10:12:33:20"
```

#### Normal Play Time

Normal Play Time expresses time in terms of SMIL clock values. The metric specifier is "npt", and the syntax of the time value is identical to the syntax of SMIL clock values.

Examples:

```
clip-begin="npt=123.45s"
```

```
clip-begin="npt=12:05:35.3"
```

#### clip-end

The clip-end attribute specifies the end of a sub-clip of a continuous media object (such as audio, video or another presentation) that should be played. It uses the same attribute value syntax as the clip-begin attribute.

If the value of the "clip-end" attribute exceeds the duration of the media object, the value is ignored, and the clip end is set equal to the effective end of the media object.

#### copyright

Defined in [Section 4.2.1](#)

#### dur

Defined in [Section 4.2.1](#)

#### end

Defined in [Section 4.2.1](#)

#### fill

For a definition of the semantics of this attribute, see Section 4.2.4. The attribute can have the values "remove" and "freeze".

#### id

Defined in [Section 2](#)

#### longdesc

This attribute specifies a link (URI) to a long description of a media object. This description should supplement the short description provided using the alt attribute. When the media-object has associated anchors, this attribute should provide information about the anchor's contents.

#### region

Defined in [Section 4.2.1](#)

#### src

The value of the src attribute is the URI of the media object.

#### system-bitrate

Defined in [Section 4.4](#)

#### system-captions

Defined in [Section 4.4](#)

system-language

Defined in [Section 4.4](#)

system-overdub-or-caption

Defined in [Section 4.4](#)

system-required

Defined in [Section 4.4](#)

system-screen-size

Defined in [Section 4.4](#)

system-screen-depth

Defined in [Section 4.4](#)

title

Defined in [Section 3.3.1](#)

It is strongly recommended that all media object elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

type

MIME type of the media object referenced by the "src" attribute.

## Element Content

Media Object Elements can contain the following element:

anchor

Defined in [Section 4.5.2](#)

### 4.2.4 SMIL Time Model

#### 4.2.4.1 Time Model Values

In the following discussion, the term "element" refers to synchronization elements only.

For each element we define the implicit, explicit, desired, and effective begin, duration, and end.

The effective begin/duration/end specify what the reader of the document will perceive.

The implicit, explicit, and desired values are auxiliary values used to define the effective values.

The rules for calculating each of these values for the elements defined in SMIL 1.0 are described in the next section.

1. Each element in SMIL has an *implicit begin*.
2. Each element can be assigned an *explicit begin* by adding a "begin" attribute to the element:

$$\text{begin} = \text{"value of explicit-begin"}$$

It is an error if the explicit begin is earlier than the implicit begin of the element.

3. Each element in SMIL has an *implicit end*.
4. Each element can be assigned an *explicit end* by adding an "end" attribute to the element:

$$\text{end} = \text{"value of explicit-end"}$$

5. The *implicit duration* of an element is the difference between the implicit end and the implicit begin.
6. Each element in SMIL can be assigned an *explicit duration* by adding a "dur" attribute to the element:

`dur = "value of explicit-duration"`

7. The *desired begin* of an element is equal to the explicit begin if one is given, otherwise the desired begin is equal to the implicit begin.
8. Each element has a *desired end*.
9. The *desired duration* of an element is the difference between the desired end and the desired begin.
10. Each element has an *effective begin*.
11. Each element has an *effective end*. (Note: the effective end of a child element can never be later than the effective end of its parent.)
12. The *effective duration* of an element is the difference between the effective end and the effective begin.

#### 4.2.4.2 Determining Time Model Values for SMIL 1.0 Elements

This section defines how time model values are calculated for the synchronization elements of SMIL 1.0 in cases that are not covered by the rules in [Section 4.2.4.1](#).

##### Determining the *implicit begin* of an element

- The implicit begin of the first child of the "body" element is when the document starts playing. When this falls outside the scope of this document.
- The implicit begin of a child of a "par" element is equal to the effective begin of the "par" element.
- The implicit begin of the first child of a "seq" element is equal to the effective begin of the "seq" element.
- The implicit begin of any other child of a "seq" element is equal to the desired end time of the previous child of the "seq" element.

##### Determining the *implicit end* of an element

The first description that matches the element is the one that is to be used:

- An element with a "repeat" attribute with value "indefinite" has an implicit end immediately after its effective begin.
- An element with a "repeat" attribute with a value other than "indefinite" has an implicit end equal to the implicit end of a seq element with the stated number of copies of the element without "repeat" attribute as children.
- A media object element referring to a continuous media object has an implicit end equal to the sum of the effective begin of the element and the intrinsic duration of the media object.
- A media object element referring to a discrete media object such as text or image has an implicit end immediately after its effective begin.
- A "seq" element has an implicit end equal to the desired end of its last child.
- A "par" element has an implicit end that depends on the value of the "endsync" attribute. The implicit end is equal to the sum of the effective begin of the "par" element and the implicit duration which is derived as follows:
  - If the value of the "endsync" attribute is "last", or if the "endsync" attribute is missing, the implicit duration of the "par" element is the maximum of the desired durations of its children.

- If the value of the "endsync" attribute is "first", the implicit duration of the "par" element is the minimum of the desired durations of its children.
- If the value of the "endsync" attribute is an id-ref, the implicit duration of the "par" element is equal to the desired duration of the child referenced by the "id-ref".

#### **Determining the *desired end* of an element**

- If the element has both an explicit duration and an explicit end, the desired end is the minimum of:
  - the sum of the desired begin and the explicit duration; and
  - the explicit end.
- If the element has an explicit duration but no explicit end, the desired end is the sum of the desired begin and the explicit duration.
- If the element has an explicit end but no explicit duration, the desired end is equal to the explicit end
- Otherwise, the desired end is equal to the implicit end.

#### **Determining the *desired begin* of an element**

The desired begin of an element is determined by using rule 7 in [Section 4.2.4.1](#).

#### **Determining the *effective begin* of an element**

The *effective begin* of an element is equal to the desired begin of the element, unless the effective end of the parent element is earlier than this time, in which case the element is not shown at all.

#### **Determining the *effective end* of an element**

- The effective end of the last child of the body element is player-dependent. The effective end is at least as late as the desired end, but whether it is any later is implementation-dependent.
- The effective end of the child of a `par` element can be derived as follows:
  - If the child has a `fill` attribute, and the value of the `fill` attribute is "freeze", the effective end of the child element is equal to the effective end of the parent. The last state of the element is retained on the screen until the effective end of the element.
  - If the child has a `fill` attribute, and the value of the `fill` attribute is "remove", the effective end of the child element is the minimum of the effective end of the parent and the desired end of the child element.
  - If the child element has no `fill` attribute, the effective end of the child depends on whether or not the child has an explicit duration or end.
    - If the child has an explicit duration or end, the effective end is determined as if the element had a `fill` attribute with value "remove".
    - If the child has neither an explicit duration nor an explicit end, the effective end is determined as if the element had a `fill` attribute with value "freeze".
- The effective end of the last child of a `seq` element is derived in the same way as the effective end of a child of a `par` element.
- The effective end of any other child of a `seq` element can be derived as follows:
  - If the child has a `fill` attribute, and the value of the `fill` attribute is "freeze", the effective end of the child element is equal to the effective begin of the next element
  - If the child has a `fill` attribute, and the value of the `fill` attribute is "remove", the effective end of the child element is the minimum of the effective begin of the next element and the desired end of the next child element.
  - If the child element has no `fill` attribute, the effective end of the child depends on whether or not the child has an explicit duration or end.

- If the child has an explicit duration or end, the effective end is determined as if the element had a fill attribute with value "remove".
- If the child has neither an explicit duration nor an explicit end, the effective end is determined as if the element had a fill attribute with value "freeze".

### 4.3 The `switch` Element

The switch element allows an author to specify a set of alternative elements from which only one acceptable element should be chosen. An element is acceptable if the element is a SMIL 1.0 element, the media-type can be decoded, and all of the test-attributes (see [Section 4.4](#)) of the element evaluate to "true".

An element is selected as follows: the player evaluates the elements in the order in which they occur in the switch element. The first acceptable element is selected at the exclusion of all other elements within the switch.

Thus, authors should order the alternatives from the most desirable to the least desirable. Furthermore, authors should place a relatively fail-safe alternative as the last item in the <switch> so that at least one item within the switch is chosen (unless this is explicitly not desired). Implementations should NOT arbitrarily pick an object within a <switch> when test-attributes for all fail.

Note that http URIs provide for content-negotiation, which may be an alternative to using the "switch" element in some cases.

#### Attributes

The switch element can have the following attributes:

<code>id</code>	Defined in <a href="#">Section 2</a>
<code>title</code>	Defined in <a href="#">Section 3.3.1</a> It is strongly recommended that all switch elements have a <code>title</code> attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

#### Element Content

If the `switch` element is used as a direct or indirect child of a `body` element, it can contain the following children:

<code>a</code>	Defined in <a href="#">Section 4.5.1</a>
<code>animation</code>	Defined in <a href="#">Section 4.2.3</a>
<code>audio</code>	Defined in <a href="#">Section 4.2.3</a>
<code>img</code>	Defined in <a href="#">Section 4.2.3</a>
<code>par</code>	Defined in <a href="#">Section 4.2.1</a>
<code>ref</code>	Defined in <a href="#">Section 4.2.3</a>
<code>seq</code>	Defined in <a href="#">Section 4.2.2</a>
<code>switch</code>	Defined in <a href="#">Section 4.3</a>
<code>text</code>	Defined in <a href="#">Section 4.2.3</a>
<code>textstream</code>	Defined in <a href="#">Section 4.2.3</a>
<code>video</code>	Defined in <a href="#">Section 4.2.3</a>

All of these elements may appear multiple times as children of a `switch` element.

If the `switch` element is used within a "head" element, it can contain the following child:

`layout` Defined in [Section 3.2](#)  
Multiple layout elements may occur within the `switch` element.

## 4.4 Test Attributes

This specification defines a list of test attributes that can be added to any synchronization element, and that test system capabilities and settings. Conceptually, these attributes represent boolean tests. When one of the test attributes specified for an element evaluates to "false", the element carrying this attribute is ignored.

Within the list below, the concept of "user preference" may show up. User preferences are usually set by the playback engine using a preferences dialog box, but this specification does not place any restrictions on how such preferences are communicated from the user to the SMIL player.

The following test attributes are defined in SMIL 1.0:

### `system-bitrate`

This attribute specifies the approximate bandwidth, in bits per second available to the system. The measurement of bandwidth is application specific, meaning that applications may use sophisticated measurement of end-to-end connectivity, or a simple static setting controlled by the user. In the latter case, this could for instance be used to make a choice based on the users connection to the network. Typical values for modem users would be 14400, 28800, 56000 bit/s etc. Evaluates to "true" if the available system bitrate is equal to or greater than the given value. Evaluates to "false" if the available system bitrate is less than the given value. The attribute can assume any integer value greater than 0. If the value exceeds an implementation-defined maximum bandwidth value, the attribute always evaluates to "false".

### `system-captions`

This attribute allows authors to distinguish between a redundant text equivalent of the audio portion of the presentation (intended for a audiences such as those with hearing disabilities or those learning to read who want or need this information) and text intended for a wide audience. The attribute can has the value "on" if the user has indicated a desire to see closed-captioning information, and it has the value "off" if the user has indicated that they don't wish to see such information. Evaluates to "true" if the value is "on", and evaluates to "false" if the value is "off".

### `system-language`

The attribute value is a comma-separated list of language names as defined in [RFC1766].

Evaluates to "true" if one of the languages indicated by user preferences exactly equals one of the languages given in the value of this parameter, or if one of the languages indicated by

user preferences exactly equals a prefix of one of the languages given in the value of this parameter such that the first tag character following the prefix is "-".

Evaluates to "false" otherwise.

Note: This use of a prefix matching rule does not imply that language tags are assigned to languages in such a way that it is always true that if a user understands a language with a certain tag, then this user will also understand all languages with tags for which this tag is a prefix.

The prefix rule simply allows the use of prefix tags if this is the case.

Implementation note: When making the choice of linguistic preference available to the user, implementors should take into account the fact that users are not familiar with the details of language matching as described above, and should provide appropriate guidance. As an example, users may assume that on selecting "en-gb", they will be served any kind of English document if British English is not available. The user interface for setting user preferences should guide the user to add "en" to get the best matching behavior.

Multiple languages MAY be listed for content that is intended for multiple audiences. For example, a rendition of the "Treaty of Waitangi", presented simultaneously in the original Maori and English versions, would call for:

```
<audio src="foo.rm" system-language="mi, en"/>
```

However, just because multiple languages are present within the object on which the system-language test attribute is placed, this does not mean that it is intended for multiple linguistic audiences. An example would be a beginner's language primer, such as "A First Lesson in Latin," which is clearly intended to be used by an English-literate audience. In this case, the system-language test attribute should only include "en".

Authoring note: Authors should realize that if several alternative language objects are enclosed in a "switch", and none of them matches, this may lead to situations such as a video being shown without any audio track. It is thus recommended to include a "catch-all" choice at the end of such a switch which is acceptable in all cases.

#### system-overdub-or-caption

This attribute is a setting which determines if users prefer overdubbing or captioning when the option is available. The attribute can have the values "caption" and "overdub". Evaluates to "true" if the user preference matches this attribute value. Evaluates to "false" if they do not match.

#### system-required

This attribute specifies the name of an extension. Evaluates to "true" if the extension is supported by the implementation, otherwise, this evaluates to "false". In a future version of SMIL, this attribute value will be an XML namespace [NAMESPACES].

#### system-screen-size

Attribute values have the following syntax:

```
screen-size-val ::= screen-height "X" screen-width
```

Each of these is a pixel value, and must be an integer value greater than 0. Evaluates to "true" if the SMIL playback engine is capable of displaying a presentation of the given size. Evaluates to "false" if the SMIL playback engine is only capable of displaying a smaller presentation.

#### system-screen-depth

This attribute specifies the depth of the screen color palette in bits required for displaying the element. The value must be greater than 0. Typical values are 1, 8, 24 .... Evaluates to "true" if the SMIL playback engine is capable of displaying images or video with the given color depth. Evaluates to "false" if the SMIL playback engine is only capable of displaying images or video with a smaller color depth.

## Examples

### 1) Choosing between content with different bitrate

In a common scenario, implementations may wish to allow for selection via a "system-bitrate" parameter on elements. The media player evaluates each of the "choices" (elements within the switch) one at a time, looking for an acceptable bitrate given the known characteristics of the link between the media player and media server.

```

...
<par>
  <text .../>
  <switch>
    <par system-bitrate="40000">
      ...
    </par>
    <par system-bitrate="24000">
      ...
    </par>
    <par system-bitrate="10000">
      .....
    </par>
  </switch>
</par>
...

```

### 2) *Choosing between audio resources with different bitrate*

The elements within the switch may be any combination of elements. For instance, one could merely be specifying an alternate audio track:

```

...
<switch>
  <audio src="joe-audio-better-quality" system-bitrate="16000" />
  <audio src="joe-audio" system-bitrate="8000" />
</switch>
...

```

### 3) *Choosing between audio resources in different languages*

In the following example, an audio resource is available both in French and in English. Based on the user's preferred language, the player can choose one of these audio resources.

```

...
<switch>
  <audio src="joe-audio-french" system-language="fr" />
  <audio src="joe-audio-english" system-language="en" />
</switch>
...

```

### 4) *Choosing between content written for different screens*

In the following example, the presentation contains alternative parts designed for screens with different resolutions and bit-depths. Depending on the particular characteristics of the screen, the player can choose one of the alternatives.

```

...
<par>
  <text .../>
  <switch>
    <par system-screen-size="1280X1024" system-screen-depth="16">
      .....
    </par>
  </switch>
</par>

```

```

    <par system-screen-size="640X480" system-screen-depth="32">
    ...
    </par>
    <par system-screen-size="640X480" system-screen-depth="16">
    ...
    </par>
  </switch>
</par>
...

```

### 5) Distinguishing caption tracks from stock tickers

In the following example, captions are shown only if the user wants captions on.

```

...
<seq>
  <par>
    <audio      src="audio.rm" />
    <video      src="video.rm" />
    <textstream src="stockticker.rtx" />
    <textstream src="closed-caps.rtx" system-captions="on" />
  </par>
</seq>
...

```

### 6) Choosing the language of overdub and caption tracks

In the following example, a French-language movie is available with English, German, and Dutch overdub and caption tracks. The following SMIL segment expresses this, and switches on the alternatives that the user prefers.

```

...
<par>
  <switch>
    <audio src="movie-aud-en.rm" system-language="en"
          system-overdub-or-caption="overdub" />
    <audio src="movie-aud-de.rm" system-language="de"
          system-overdub-or-caption="overdub" />
    <audio src="movie-aud-nl.rm" system-language="nl"
          system-overdub-or-caption="overdub" />
    <!-- French for everyone else -->
    <audio src="movie-aud-fr.rm" />
  </switch>
  <video src="movie-vid.rm" />
  <switch>
    <textstream src="movie-caps-en.rtx" system-language="en"
              system-overdub-or-caption="caption" />
    <textstream src="movie-caps-de.rtx" system-language="de"
              system-overdub-or-caption="caption" />
    <textstream src="movie-caps-nl.rtx" system-language="nl"
              system-overdub-or-caption="caption" />
    <!-- French captions for those that really want them -->
    <textstream src="movie-caps-fr.rtx" system-captions="on" />
  </switch>
</par>

```

...

## 4.5 Hyperlinking Elements

The link elements allows the description of navigational links between objects.

SMIL provides only for in-line link elements. Links are limited to uni-directional single-headed links (i.e. all links have exactly one source and one destination resource). All links in SMIL are actuated by the user.

### Handling of Links in Embedded Documents

Due to its integrating nature, the presentation of a SMIL document may involve other (non-SMIL) applications or plug-ins. For example, a SMIL browser may use an HTML plug-in to display an embedded HTML page. Vice versa, an HTML browser may use a SMIL plug-in to display a SMIL document embedded in an HTML page.

In such presentations, links may be defined by documents at different levels and conflicts may arise. In this case, the link defined by the containing document should take precedence over the link defined by the embedded object. Note that since this might require communication between the browser and the plug-in, SMIL implementations may choose not to comply with this recommendation.

If a link is defined in an embedded SMIL document, traversal of the link affects only the embedded SMIL document.

If a link is defined in a non-SMIL document which is embedded in a SMIL document, link traversal can only affect the presentation of the embedded document and not the presentation of the containing SMIL document. This restriction may be released in future versions of SMIL.

### Addressing

SMIL supports name fragment identifiers and the '#' connector. This means that SMIL supports locators as currently used in HTML (e.g. it uses locators of the form "http://foo.com/some/path#anchor1").

### Linking to SMIL Fragments

A locator that points to a SMIL document may contain a fragment part (e.g. http://www.w3.org/test.smi#par1). The fragment part is an id value that identifies one of the elements within the referenced SMIL document. If a link containing a fragment part is followed, the presentation should start as if the user had fast-forwarded the presentation represented by the destination document to the effective begin of the element designated by the fragment.

The following special cases can occur:

1. The element addressed by the link has a "repeat" attribute.
  1. If the value of the "repeat" attribute is N, all N repetitions of the element are played.
  2. If the value of the "repeat" attribute is "indefinite", playback ends according to the rules defined for repeat value "indefinite".
2. The element addressed by the link is contained within another element that contains a "repeat" attribute.
  1. If the value of the "repeat" attribute is N, playback starts at the beginning of the element addressed by the link, followed by N-1 repetitions of the element containing the "repeat" attribute.

2. If the value of the "repeat" attribute is "indefinite", playback starts at the beginning of the element addressed by the link. Playback ends according to the rules defined for repeat value "indefinite".
3. The element addressed by the link is content of a "switch" element: It is forbidden to link to elements that are the content of "switch" elements.

#### 4.5.1 The a Element

The functionality of the "a" element is very similar to the functionality of the "a" element in HTML 4.0 [\[HTML40\]](#). SMIL adds an attribute "show" that controls the temporal behavior of the source when the link is followed. For synchronization purposes, the "a" element is transparent, i.e. it does not influence the synchronization of its child elements. "a" elements may not be nested. An "a" element must have an [href](#) attribute.

##### Attributes

The a element can have the following attributes:

id

Defined in [Section 2](#)

href

This attribute contains the URI of the link's destination.  
The "href" attribute is required for "a" elements.

[show](#)

This attribute controls the behavior of the source document containing the link when the link is followed. It can have one of the following values:

- "replace": The current presentation is paused at its current state and is replaced by the destination resource. If the player offers a history mechanism, the source presentation resumes from the state in which it was paused when the user returns to it.
- "new": The presentation of the destination resource starts in a new context, not affecting the source resource.
- "pause": The source presentation is paused at its current state, and the destination resource starts in a new context. When the display of the destination resource ends, the source presentation resumes from the state in which it was paused.

The default value of "show" is "replace".

title

Defined in [Section 3.3.1](#)

It is strongly recommended that all "a" elements have a title attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

##### Element Content

The a element can contain the following children:

animation

Defined in [Section 4.2.3](#)

audio

Defined in [Section 4.2.3](#)

img

Defined in [Section 4.2.3](#)

par

Defined in [Section 4.2.1](#)

ref

Defined in [Section 4.2.3](#)

seq	Defined in <a href="#">Section 4.2.2</a>
switch	Defined in <a href="#">Section 4.3</a>
text	Defined in <a href="#">Section 4.2.3</a>
textstream	Defined in <a href="#">Section 4.2.3</a>
video	Defined in <a href="#">Section 4.2.3</a>

## Examples

### *Example 1*

The link starts up the new presentation replacing the presentation that was playing.

```
<a href="http://www.cwi.nl/somewhereelse.smi">
  <video src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

In the example, the second line can be replaced by a reference to any valid subtree of an SMIL presentation.

### *Example 2*

The link starts up the new presentation in addition to the presentation that was playing.

```
<a href="http://www.cwi.nl/somewhereelse.smi" show="new">
  <video src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

For example, this allows a SMIL player to spawn off an HTML browser.

### *Example 3*

The link starts up the new presentation and pauses the presentation that was playing.

```
<a href="http://www.cwi.nl/somewhereelse.smi" show="pause">
  <video src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

### *Example 4*

The following example contains a link from an element in one presentation A to the middle of another presentation B. This would play presentation B starting from the effective begin of the element with id "next".

Presentation A:

```
<a href="http://www.cwi.nl/presentationB#next">
  <video src="rtsp://foo.com/graph.imf"/>
</a>
```

Presentation B (<http://www.cwi.nl/presentation>):

...

```

<seq>
  <video src="rtsp://foo.com/graph.imf" />
  <par>
    <video src="rtsp://foo.com/timbl.rm" region="l_window" />
    <video id="next" src="rtsp://foo.com/v1.rm" region="r_window" />
      ^^^^^^^^^
    <text src="rtsp://foo.com/caption1.html" region="l_2_title" />
    <text src="rtsp://foo.com/caption2.rtx" region="r_2_title" />
  </par>
</seq>
...

```

#### 4.5.2 The anchor Element

The functionality of the "a" element is restricted in that it only allows associating a link with a complete media object. HTML image maps have demonstrated that it is useful to associate links with spatial subparts of an object. The anchor element realizes similar functionality for SMIL:

1. The anchor element allows associating a link destination to spatial and temporal subparts of a media object, using the "href" attribute (in contrast, the "a" element only allows associating a link with a complete media object).
2. The anchor element allows making a subpart of the media object the destination of a link, using the "id" attribute.
3. The anchor element allows breaking up an object into spatial subparts, using the "coords" attribute.
4. The anchor element allows breaking up an object into temporal subparts, using the "begin" and "end" attributes. The values of the begin and end attributes are relative to the beginning of the media object.

#### Attributes

The anchor element can have the following attributes:

begin Defined in [Section 4.2.1](#)

#### coords

The value of this attribute specifies a rectangle within the display area of a visual media object. Syntax and semantics of this attribute are similar to the coords attribute in HTML image maps, when the link is associated with a rectangular shape. The rectangle is specified by four length values: The first two values specify the coordinates of the upper left corner of the rectangle. The second two values specify the coordinates of the lower right corner of the rectangle. Coordinates are relative to the top left corner of the visual media object (see Figure 4.5). If a coordinate is specified as a percentage value, it is relative to the total width or height of the media object display area.

An attribute with an erroneous coords value is ignored (right-x smaller or equal to left-x, bottom-y smaller or equal to top-y). If the rectangle defined by the coords attribute exceeds the area covered by the media object, exceeding height and width are clipped at the borders of the media object.

Values of the coords attribute have the following syntax:

```
coords-value ::= left-x "," top-y "," right-x "," bottom-y
```

---

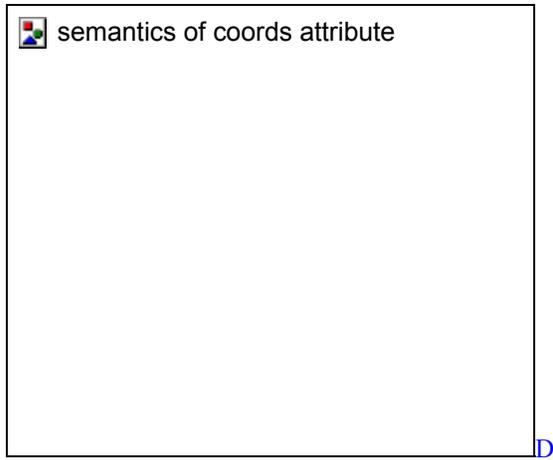


Figure 4.5: Semantics of "coords" attribute

end	Defined in <a href="#">Section 4.2.1</a>
id	Defined in <a href="#">Section 2</a>
show	Defined in <a href="#">Section 4.5.1</a>
skip-content	Defined in <a href="#">Section 3.3.1</a>
title	Defined in <a href="#">Section 3.3.1</a>

It is strongly recommended that all anchor elements have a `title` attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

## Examples

### 1) Associating links with spatial subparts

In the following example, the screenspace taken up by a video clip is split into two sections. A different link is associated with each of these sections.

```
<video src="http://www.w3.org/CoolStuff">
  <anchor href="http://www.w3.org/AudioVideo" coords="0%,0%,50%,50%"/>
  <anchor href="http://www.w3.org/Style"
coords="50%,50%,100%,100%"/>
</video>
```

### 2) Associating links with temporal subparts

In the following example, the duration of a video clip is split into two subintervals. A different link is associated with each of these subintervals.

```
<video src="http://www.w3.org/CoolStuff">
  <anchor href="http://www.w3.org/AudioVideo" begin="0s" end="5s"/>
  <anchor href="http://www.w3.org/Style" begin="5s" end="10s"/>
</video>
```

### 3) *Jumping to a subpart of an object*

The following example contains a link from an element in one presentation A to the middle of a video object contained in another presentation B. This would play presentation B starting from second 5 in the video (i.e. the presentation would start as if the user had fast-forwarded the whole presentation to the point at which the designated fragment in the "CoolStuff" video begins).

Presentation A:

```
<a href="http://www.cwi.nl/mm/presentationB#tim">
  <video id="graph" src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

Presentation B:

```
<video src="http://www.w3.org/CoolStuff">
  <anchor id="joe" begin="0s" end="5s"/>
  <anchor id="tim" begin="5s" end="10s"/>
</video>
```

### 4) *Combining different uses of links*

The following example shows how the different uses of associated links can be used in combination.

Presentation A:

```
<a href="http://www.cwi.nl/mm/presentationB#tim">
  <video id="graph" src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

Presentation B:

```
<video src="http://www.w3.org/CoolStuff">
  <anchor id="joe" begin="0s" end="5s" coords="0%,0%,50%,50%"
    href="http://www.w3.org/" />
  <anchor id="tim" begin="5s" end="10s" coords="0%,0%,50%,50%"
    href="http://www.w3.org/Tim" />
</video>
```

## 5 SMIL DTD

### 5.1 Relation to XML

A SMIL 1.0 document may optionally contain a document type declaration, which names the document type definition (DTD) in use for the document. For SMIL, the document type declaration should look as follows (the double quotes can be replaced by single quotes):

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"
    "http://www.w3.org/TR/REC-smil/SMIL10.dtd">
```

The XML 1.0 specification provides a way to extend the DTD using the <!DOCTYPE> element, for instance to add a new set of entity definitions. Authors must not use this feature with SMIL since many SMIL players will not support it.

The following is illegal in SMIL:

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"
    "http://www.w3.org/TR/REC-smil/SMIL10.dtd" [
<!ENTITY % AcmeCorpSymbols PUBLIC
    "-//Acme Corp//ENTITIES Corporate Symbols//EN"
    "http://www.acme.com/corp_symbols.xml"
>
%AcmeCorpSymbols;
]>
```

### 5.2 DTD

```
<!--
```

This is the XML document type definition (DTD) for SMIL 1.0.

Date: 1998/06/15 08:56:30

Authors:

Jacco van Ossenbruggen <jrvosse@cw.nl>  
Sjoerd Mullender <sjoerd@cw.nl>

Further information about SMIL is available at:

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

```
-->
```

```
<!-- Generally useful entities -->
<!ENTITY % id-attr "id ID #IMPLIED">
<!ENTITY % title-attr "title CDATA #IMPLIED">
<!ENTITY % skip-attr "skip-content (true|false) 'true' ">
<!ENTITY % desc-attr "
    %title-attr;
    abstract          CDATA #IMPLIED
    author            CDATA #IMPLIED
```

```

        copyright      CDATA      #IMPLIED
">

<!--===== SMIL Document =====>
<!--
    The root element SMIL contains all other elements.
-->
<!ELEMENT smil (head?,body?)>
<!ATTLIST smil
        %id-attr;
>

<!--===== The Document Head =====>
<!ENTITY % layout-section "layout|switch">

<!ENTITY % head-element "(meta*,((%layout-section;), meta*))?">

<!ELEMENT head %head-element;>
<!ATTLIST head %id-attr;>

<!--===== Layout Element =====>
<!--
    Layout contains the region and root-layout elements defined by
    smil-basic-layout or other elements defined an external layout
    mechanism.
-->
<!ELEMENT layout ANY>
<!ATTLIST layout
        %id-attr;
        type CDATA      "text/smil-basic-layout"
>

<!--===== Region Element =====>
<!ENTITY % viewport-attrs "
        height      CDATA      #IMPLIED
        width       CDATA      #IMPLIED
        background-color CDATA      #IMPLIED
">

<!ELEMENT region EMPTY>
<!ATTLIST region
        %id-attr;
        %title-attr;
        %viewport-attrs;
        left        CDATA      "0"
        top         CDATA      "0"
        z-index     CDATA      "0"
        fit         (hidden|fill|meet|scroll|slice)      "hidden"
        %skip-attr;
>

<!--===== Root-layout Element =====>
<!ELEMENT root-layout EMPTY>
<!ATTLIST root-layout
        %id-attr;

```

```

        %title-attr;
        %viewport-attrs;
        %skip-attr;
    >

<!--===== Meta Element =====>
<!ELEMENT meta EMPTY>
<!ATTLIST meta
    name      NMTOKEN #REQUIRED
    content   CDATA   #REQUIRED
    %skip-attr;
>

<!--===== The Document Body =====>
<!ENTITY % media-object
"audio|video|text|img|animation|textstream|ref">
<!ENTITY % schedule "par|seq|(%media-object;)">
<!ENTITY % inline-link "a">
<!ENTITY % assoc-link "anchor">
<!ENTITY % link "%inline-link;">
<!ENTITY % container-content "(%schedule;)|switch|(%link;)">
<!ENTITY % body-content "(%container-content;)">

<!ELEMENT body (%body-content;)*>
<!ATTLIST body %id-attr;>

<!--===== Synchronization Attributes =====>
<!ENTITY % sync-attributes "
    begin   CDATA   #IMPLIED
    end     CDATA   #IMPLIED
">

<!--===== Switch Parameter Attributes =====>
<!ENTITY % system-attribute "
    system-bitrate           CDATA           #IMPLIED
    system-language         CDATA           #IMPLIED
    system-required         NMTOKEN         #IMPLIED
    system-screen-size      CDATA           #IMPLIED
    system-screen-depth     CDATA           #IMPLIED
    system-captions         (on|off)        #IMPLIED
    system-overdub-or-caption (caption|overdub) #IMPLIED
">

<!--===== Fill Attribute =====>
<!ENTITY % fill-attribute "
    fill      (remove|freeze)    'remove'
">

<!--===== The Parallel Element =====>
<!ENTITY % par-content "%container-content;">
<!ELEMENT par      (%par-content;)*>
<!ATTLIST par
    %id-attr;
    %desc-attr;
    endsync CDATA           "last"

```

```

        dur      CDATA      #IMPLIED
        repeat   CDATA      "1"
        region   IDREF      #IMPLIED
        %sync-attributes;
        %system-attribute;
    >

<!--===== The Sequential Element =====>
<!ENTITY % seq-content "%container-content;">
<!ELEMENT seq      (%seq-content;)*>
<!ATTLIST seq
        %id-attr;
        %desc-attr;
        dur      CDATA      #IMPLIED
        repeat   CDATA      "1"
        region   IDREF      #IMPLIED
        %sync-attributes;
        %system-attribute;
    >

<!--===== The Switch Element =====>
<!-- In the head, a switch may contain only layout elements,
      in the body, only container elements. However, this
      constraint cannot be expressed in the DTD (?), so
      we allow both:
-->
<!ENTITY % switch-content "layout|(%container-content;)">
<!ELEMENT switch (%switch-content;)*>
<!ATTLIST switch
        %id-attr;
        %title-attr;
    >

<!--===== Media Object Elements =====>
<!-- SMIL only defines the structure. The real media data is
      referenced by the src attribute of the media objects.
-->

<!-- Furthermore, they have the following attributes as defined
      in the SMIL specification:
-->
<!ENTITY % mo-attributes "
        %id-attr;
        %desc-attr;
        region   IDREF      #IMPLIED
        alt      CDATA      #IMPLIED
        longdesc CDATA      #IMPLIED
        src      CDATA      #IMPLIED
        type     CDATA      #IMPLIED
        dur      CDATA      #IMPLIED
        repeat   CDATA      '1'
        %fill-attribute;
        %sync-attributes;
        %system-attribute;
">

<!--

```

Most info is in the attributes, media objects are empty or contain associated link elements:

```
-->
<!ENTITY % mo-content "(%assoc-link;)*">
<!ENTITY % clip-attrs "
    clip-begin      CDATA    #IMPLIED
    clip-end        CDATA    #IMPLIED
">

<!ELEMENT ref      %mo-content;>
<!ELEMENT audio   %mo-content;>
<!ELEMENT img     %mo-content;>
<!ELEMENT video   %mo-content;>
<!ELEMENT text    %mo-content;>
<!ELEMENT textstream %mo-content;>
<!ELEMENT animation %mo-content;>

<!ATTLIST ref      %mo-attributes; %clip-attrs;>
<!ATTLIST audio   %mo-attributes; %clip-attrs;>
<!ATTLIST video   %mo-attributes; %clip-attrs;>
<!ATTLIST animation %mo-attributes; %clip-attrs;>
<!ATTLIST textstream %mo-attributes; %clip-attrs;>
<!ATTLIST text    %mo-attributes;>
<!ATTLIST img     %mo-attributes;>

<!--===== Link Elements =====>

<!ENTITY % smil-link-attributes "
    %id-attr;
    %title-attr;
    href      CDATA          #REQUIRED
    show      (replace|new|pause) 'replace'
">

<!--===== Inline Link Element =====>
<!ELEMENT a (%schedule;|switch)*>
<!ATTLIST a
    %smil-link-attributes;
>

<!--===== Associated Link Element =====>
<!ELEMENT anchor EMPTY>
<!ATTLIST anchor
    %skip-attr;
    %smil-link-attributes;
    %sync-attributes;
    coords      CDATA          #IMPLIED
>
```

## References

- [CSS2] "Cascading Style Sheets, level 2", B. Bos, H. Lie, C. Lilley, I. Jacobs, 12 May 1998.  
Available at <http://www.w3.org/TR/REC-CSS2/>.
- [HTML40] "HTML 4.0 Specification", D. Raggett, A. Le Hors, I. Jacobs, 24 April 1998.  
Available at <http://www.w3.org/TR/REC-html40>.
- [ISO/IEC 10646] ISO (International Organization for Standardization). ISO/IEC 10646-1993 (E). Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane. [Geneva]: International Organization for Standardization, 1993 (plus amendments AM 1 through AM 7).
- [NAMESPACES] "Namespaces in XML", T. Bray, D. Hollander, A. Layman, 27 March 1998  
W3C working draft. Available at <http://www.w3.org/TR/WD-xml-names>.
- [PICS] "PICS 1.1 Label Distribution -- Label Syntax and Communication Protocols", 31 October 1996, T. Krauskopf, J. Miller, P. Resnick, W. Trees  
Available at <http://www.w3.org/TR/REC-PICS-labels-961031>
- [RFC1738] "Uniform Resource Locators", T. Berners-Lee, L. Masinter, and M. McCahill, December 1994.  
Available at <ftp://ftp.isi.edu/in-notes/rfc1738.txt>.
- [RFC1766] "Tags for the Identification of Languages", H. Alvestrand, March 1995.  
Available at <ftp://ftp.isi.edu/in-notes/rfc1766.txt>.
- [RFC1808] "Relative Uniform Resource Locators", R. Fielding, June 1995.  
Available at <ftp://ftp.isi.edu/in-notes/rfc1808.txt>.
- [RFC2045] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed and N. Borenstein, November 1996.  
Available at <ftp://ftp.isi.edu/in-notes/rfc2045.txt>. Note that this RFC obsoletes RFC1521, RFC1522, and RFC1590.
- [SMPTE] "Time and Control Codes for 24, 25 or 30 Frame-Per-Second Motion-Picture Systems - RP 136-1995". Society of Motion Picture & Television Engineers.
- [URI] "Uniform Resource Identifiers (URI): Generic Syntax and Semantics", T. Berners-Lee, R. Fielding, L. Masinter, 4 March 1998.  
Available at <http://www.ics.uci.edu/pub/ietf/uri/draft-fielding-uri-syntax-02.txt>. This is a work in progress that is expected to update [\[RFC1738\]](#) and [\[RFC1808\]](#).
- [XML10] "Extensible Markup Language (XML) 1.0", T. Bray, J. Paoli, C.M. Sperberg-McQueen, editors, 10 February 1998.  
Available at <http://www.w3.org/TR/REC-xml>

## Appendix

### Extending SMIL 1.0

*(non-normative)*

In the future, SMIL 1.0 may be extended by another W3C recommendation, or by private extensions.

For these extensions, it is recommended that the following rules are obeyed:

- All elements introduced in extensions must have a "skip-content" attribute (defined in Section 3.3.1) if it should be possible that their content is processed by SMIL 1.0 players.
- Private extensions must be introduced using the syntax of the XML namespace specification.

It is recommended that SMIL 1.0 players are prepared to handle documents that contain extension that obey these two rules.

Extensions should be handled using an XML namespace mechanism, once such a mechanism becomes a W3C recommendation. In the rest of the section, the syntax and semantics for XML namespaces defined in the W3C note [NAMESPACE] will be used for demonstration purposes only.

The following cases can occur:

1. The document contains a namespace declaration for the SMIL 1.0 specification that defines an empty prefix. In this case, non-SMIL 1.0 elements and attributes are only allowed in a document if they are declared using an XML namespace. The document may not contain a document type declaration for SMIL 1.0. If it does, it is invalid.

In the following example, the element "new:a" is a legal extension. The elements "mytags:a" and "b" are syntax errors, since they are not declared using an XML namespace.

2. 

```
<?xml:namespace ns="http://www.acme.com/new-smil" prefix="new"
?>
```
3. 

```
<?xml:namespace ns="http://www.w3.org/TR/PR-smil" ?>
```
4. 

```
<smil>
```
5. 

```
  <body>
```
6. 

```
    <par>
```
7. 

```
      <new:a>
```
8. 

```
      ...
```
9. 

```
    </new:a>
```
10. 

```
    <mytags:a ... />
```
11. 

```
    ...
```
12. 

```
  </mytags:a>
```
13. 

```
    <b>
```
14. 

```
    ...
```
15. 

```
  </b>
```
16. 

```
</par>
```
17. 

```
</body>
```
18. 

```
</smil>
```

18. The document contains no document type declaration, it contains a document type declaration for a SMIL version higher than 1.0, or it contains a namespace declaration for a SMIL specification with a version higher than 1.0. For a SMIL 1.0 player to be able to recognize such a namespace declaration, it is recommended that the URI of future SMIL versions starts with <http://www.w3.org/TR/REC-smil>, and is followed by more characters which may for example be

a version number.

In this case, a SMIL 1.0 player should assume that it is processing a SMIL document with a version number higher than 1.0.

The following cases can occur:

Unknown element

Unknown elements are ignored

An unknown element may contain content that consists of SMIL 1.0 elements. Whether such content is ignored or processed depends on the value of the "skip-content" attribute. If the attribute is set to "true", or the attribute is absent, the content is not processed. If it is set to "false", the content is processed.

Content in Element that was declared "Empty"

A future version of SMIL may allow content in elements that are declared as "empty" in SMIL 1.0.

Whether this content is ignored or not depends on the value of the "skip-content" attribute of the formerly empty element. If the attribute is set to "true", the content is not processed. If it is set to "false", the content is processed.

Unknown Attribute

Unknown attributes are ignored.

Unknown Attribute Value

Attributes with unknown attribute values are ignored.

19. The document contains a document type declaration for SMIL 1.0. In this case, it may not contain any non-SMIL 1.0 elements, even if they are declared using XML namespaces. This is because such extensions would render the document invalid.

## Using SMIL 1.0 as an Extension

When the XML namespace mechanism is used to include SMIL elements and attributes in other XML-based documents, it is recommended to use the following namespace identifier:

`http://www.w3.org/TR/REC-smil`

## Registered Media Types

[RFC2045,RFC2046] specifies that Content Types, Content Subtypes, CharacterSets, Access Types, and conversion values for MIME mail will be assigned and listed by the IANA.

### Content Types and Subtypes

<u>Type</u>	<u>Subtype</u>	<u>Description</u>	<u>Reference</u>
<b>text</b>	plain		[RFC2646,RFC2046]
	richtext		[RFC2045,RFC2046]
	enriched		[RFC1896]
	tab-separated-values		[Paul Lindner]
	html		[RFC2854]
	sgml		[RFC1874]
	vnd.latex-z		[Lubos]
	vnd.fmi.flexstor		[Hurtta]
	uri-list		[RFC2483]
	vnd.abc		[Allen]
	rfc822-headers		[RFC1892]
	vnd.in3d.3dml		[Powers]
	prs.lines.tag		[Lines]
	vnd.in3d.spot		[Powers]
	css		[RFC2318]
	xml		[RFC2376]
	rtf		[Lindner]
	directory		[RFC2425]
	calendar		[RFC2445]
	vnd.wap.wml		[Stark]
	vnd.wap.wmlscript		[Stark]
	vnd.motorola.reflex		[Patton]
	vnd.fly		[Gurney]
	vnd.wap.sl		[WAP-Forum]
	vnd.wap.si		[WAP-Forum]
	t140		[RFC2793]
	vnd.ms-mediapackage		[Nelson]
	vnd.IPTC.NewsML		[IPTC]
	vnd.IPTC.NITF		[IPTC]
	vnd.curl		[Hodge]
	vnd.DMClientScript		[Bradley]
	parityfec		[RFC3009]
<b>multipart</b>	mixed		[RFC2045,RFC2046]
	alternative		[RFC2045,RFC2046]
	digest		[RFC2045,RFC2046]
	parallel		[RFC2045,RFC2046]
	appledouble		[MacMime,Patrik Faltstrom]
	header-set		[Dave Crocker]
	form-data		[RFC2388]
	related		[RFC2387]
	report		[RFC1892]
	voice-message		[RFC2421,RFC2423]
	signed		[RFC1847]
	encrypted		[RFC1847]
	byteranges		[RFC2068]

<u>Type</u>	<u>Subtype</u>	<u>Description</u>	<u>Reference</u>	
<b>message</b>	rfc822		[RFC2045,RFC2046]	
	partial		[RFC2045,RFC2046]	
	external-body		[RFC2045,RFC2046]	
	news	[RFC 1036, Henry Spencer]		
	http		[RFC2616]	
	delivery-status		[RFC1894]	
	disposition-notification		[RFC2298]	
	s-http		[RFC2660]	
	<b>application</b>	octet-stream		[RFC2045,RFC2046]
		postscript		[RFC2045,RFC2046]
oda			[RFC2045,RFC2046]	
atomicmail		[atomicmail,Borenstein]		
andrew-inset		[andrew-inset,Borenstein]		
slate		[slate,terry crowley]		
wita		[Wang Info Transfer,Larry Campbell]		
dec-dx		[Digital Doc Trans, Larry Campbell]		
dca-rft		[IBM Doc Content Arch, Larry Campbell]		
activemessage		[Ehud Shapiro]		
rtf		[Paul Lindner]		
applefile		[MacMime,Patrik Faltstrom]		
mac-binhex40		[MacMime,Patrik Faltstrom]		
news-message-id		[RFC1036, Henry Spencer]		
news-transmission		[RFC1036, Henry Spencer]		
wordperfect5.1		[Paul Lindner]		
pdf		[Paul Lindner]		
zip		[Paul Lindner]		
macwriteii		[Paul Lindner]		
mword		[Paul Lindner]		
remote-printing		[RFC1486,Rose]		
mathematica		[Van Nostern]		
cybercash		[Eastlake]		
commonground		[Glazer]		
iges		[Parks]		
riscos		[Smith]		
eshop		[Katz]		
x400-bp		[RFC1494]		
sgml		[RFC1874]		
cals-1840		[RFC1895]		
pgp-encrypted		[RFC2015]		
pgp-signature		[RFC2015]		
pgp-keys		[RFC2015]		
vnd.framemaker		[Wexler]		
vnd.mif		[Wexler]		
vnd.ms-excel		[Gill]		
vnd.ms-powerpoint		[Gill]		
vnd.ms-project		[Gill]		
vnd.ms-works		[Gill]		
vnd.ms-tnef		[Gill]		
vnd.svd		[Becker]		
vnd.music-niff		[Butler]		
vnd.ms-artgalry		[Slawson]		
vnd.truedoc		[Chase]		
vnd.koan		[Cole]		

vnd.street-stream	[Levitt]
vnd.fdf	[Zilles]
set-payment-initiation	[Korver]
set-payment	[Korver]
set-registration-initiation	[Korver]
set-registration	[Korver]
vnd.seemail	[Webb]
vnd.businessobjects	[Imoucha]
vnd.meridian-slideshow	[Wedel]
vnd.xara	[Matthewman]
sgml-open-catalog	[Grosso]
vnd.rapid	[Szekely]
vnd.enliven	[Santinelli]
vnd.japannet-registration-wakeup	[Fujii]
vnd.japannet-verification-wakeup	[Fujii]
vnd.japannet-payment-wakeup	[Fujii]
vnd.japannet-directory-service	[Fujii]
vnd.intertrust.digibox	[Tomasello]
vnd.intertrust.nncp	[Tomasello]
prs.alvestrand.titraw-sheet	[Alvestrand]
vnd.noblenet-web	[Solomon]
vnd.noblenet-sealer	[Solomon]
vnd.noblenet-directory	[Solomon]
prs.nprend	[Doggett]
vnd.webturbo	[Rehem]
hyperstudio	[Domino]
vnd.shana.informed.formtemplate	[Selzler]
vnd.shana.informed.formdata	[Selzler]
vnd.shana.informed.package	[Selzler]
vnd.shana.informed.interchange	[Selzler]
vnd.\$commerce_battelle	[Applebaum]
vnd.osa.netdeploy	[Klos]
vnd.ibm.Minipay	[Herzberg]
vnd.japannet-jpnstore-wakeup	[Yoshitake]
vnd.japannet-setstore-wakeup	[Yoshitake]
vnd.japannet-verification	[Yoshitake]
vnd.japannet-registration	[Yoshitake]
vnd.hp-HPGL	[Pentecost]
vnd.hp-PCL	[Pentecost]
vnd.hp-PCLXL	[Pentecost]
vnd.musician	[Adams]
vnd.FloGraphIt	[Floersch]
vnd.intercon.formnet	[Gurak]
vemmi	[RFC2122]
vnd.ms-asf	[Fleischman]
vnd.ecdis-update	[Buettgenbach]
vnd.powerbuilder6	[Guy]
vnd.powerbuilder6-s	[Guy]
vnd.lotus-wordpro	[Wattenberger]
vnd.lotus-approach	[Wattenberger]
vnd.lotus-1-2-3	[Wattenberger]
vnd.lotus-organizer	[Wattenberger]
vnd.lotus-screencam	[Wattenberger]
vnd.lotus-freelance	[Wattenberger]
vnd.fujitsu.oasys	[Togashi]
vnd.fujitsu.oasys2	[Togashi]
vnd.swiftview-ics	[Widener]

vnd.dna	[Searcy]
prs.cww	[Rungchavalnont]
vnd.wt.stf	[Wohler]
vnd.dxr	[Duffy]
vnd.mitsubishi.misty-guard.trustweb	[Tanaka]
vnd.ibm.modcap	[Hohensee]
vnd.acucobol	[Lubin]
vnd.fujitsu.oasys3	[Okudaira]
marc	[RFC2220]
vnd.fujitsu.oasysprs	[Ogita]
vnd.fujitsu.oasysgp	[Sugimoto]
vnd.visio	[Sandal]
vnd.netfpx	[Mutz]
vnd.audiograph	[Slusanschi]
vnd.epson.salt	[Nagatomo]
vnd.3M.Post-it-Notes	[O'Brien]
vnd.novadigm.EDX	[Swenson]
vnd.novadigm.EXT	[Swenson]
vnd.novadigm.EDM	[Swenson]
vnd.claymore	[Simpson]
vnd.comsocaller	[Dellutri]
pkcs7-mime	[RFC2311]
pkcs7-signature	[RFC2311]
pkcs10	[RFC2311]
vnd.yellowriver-custom-menu	[Yellow]
vnd.ecowin.chart	[Olsson]
vnd.ecowin.series	[Olsson]
vnd.ecowin.filerequest	[Olsson]
vnd.ecowin.fileupdate	[Olsson]
vnd.ecowin.seriesrequest	[Olsson]
vnd.ecowin.seriesupdate	[Olsson]
EDIFACT	[RFC1767]
EDI-X12	[RFC1767]
EDI-Consent	[RFC1767]
vnd.wrq-hp3000-labelled	[Bartram]
vnd.minisoft-hp3000-save	[Bartram]
vnd.ffsns	[Holstage]
vnd.hp-hps	[Aubrey]
vnd.fujixerox.docuworks	[Taguchi]
xml	[RFC2376]
vnd.anser-web-funds-transfer-initiation	[Mori]
vnd.anser-web-certificate-issue-initiation	[Mori]
vnd.is-xpr	[Natarajan]
vnd.intu.qbo	[Scratchley]
vnd.publishare-delta-tree	[Ben-Kiki]
vnd.cybank	[Helmee]
batch-SMTP	[RFC2442]
vnd.uplanet.alert	[Martin]
vnd.uplanet.cacheop	[Martin]
vnd.uplanet.list	[Martin]
vnd.uplanet.listcmd	[Martin]
vnd.uplanet.channel	[Martin]
vnd.uplanet.bearer-choice	[Martin]
vnd.uplanet.signal	[Martin]
vnd.uplanet.alert-wbxml	[Martin]
vnd.uplanet.cacheop-wbxml	[Martin]
vnd.uplanet.list-wbxml	[Martin]

vnd.uplanet.listcmd-wbxml	[Martin]
vnd.uplanet.channel-wbxml	[Martin]
vnd.uplanet.bearer-choice-wbxml	[Martin]
vnd.epson.quickanime	[Gu]
vnd.commonspace	[Chandhok]
vnd.fut-misnet	[Pruulmann]
vnd.xfdl	[Manning]
vnd.intu.qfx	[Scratchley]
vnd.epson.ssf	[Hoshina]
vnd.epson.msf	[Hoshina]
vnd.powerbuilder7	[Shilts]
vnd.powerbuilder7-s	[Shilts]
vnd.lotus-notes	[Laramie]
pkixcmp	[RFC2510]
vnd.wap.wmlc	[Stark]
vnd.wap.wmlscriptc	[Stark]
vnd.motorola.flexsuite	[Patton]
vnd.wap.wbxml	[Stark]
vnd.motorola.flexsuite.wem	[Patton]
vnd.motorola.flexsuite.kmr	[Patton]
vnd.motorola.flexsuite.adsi	[Patton]
vnd.motorola.flexsuite.fis	[Patton]
vnd.motorola.flexsuite.gotap	[Patton]
vnd.motorola.flexsuite.ttc	[Patton]
vnd.ufdl	[Manning]
vnd.accpac.simply.imp	[Leow]
vnd.accpac.simply.aso	[Leow]
vnd.vcx	[T.Sugimoto]
ipp	[RFC2910]
ocsp-request	[RFC2560]
ocsp-response	[RFC2560]
vnd.previewsystems.box	[Smolgovsky]
vnd.mediastation.cdkey	[Flurry]
vnd.pg.format	[Gandert]
vnd.pg.osasli	[Gandert]
vnd.hp-hpid	[Gupta]
pkix-cert	[RFC2585]
pkix-crl	[RFC2585]
vnd.Mobius.TXF	[Kabayama]
vnd.Mobius.PLC	[Kabayama]
vnd.Mobius.DIS	[Kabayama]
vnd.Mobius.DAF	[Kabayama]
vnd.Mobius.MSL	[Kabayama]
vnd.cups-raster	[Sweet]
vnd.cups-postscript	[Sweet]
vnd.cups-raw	[Sweet]
index	[RFC2652]
index.cmd	[RFC2652]
index.response	[RFC2652]
index.obj	[RFC2652]
index.vnd	[RFC2652]
vnd.triscape.mxs	[Simonoff]
vnd.powerbuilder75	[Shilts]
vnd.powerbuilder75-s	[Shilts]
vnd.dpgraph	[Parker]
http	[RFC2616]
sdp	[RFC2327]

vnd.eudora.data	[Resnick]
vnd.fujixerox.docuworks.binder	[Matsumoto]
vnd.vectorworks	[Pharr]
vnd.grafeq	[Tupper]
vnd.bmi	[Gotoh]
vnd.ericsson.quickcall	[Tidwell]
vnd.hzn-3d-crossword	[Minnis]
vnd.wap.slc	[WAP-Forum]
vnd.wap.sic	[WAP-Forum]
vnd.groove-injector	[Joseph]
vnd.fujixerox.ddd	[Onda]
vnd.groove-account	[Joseph]
vnd.groove-identity-message	[Joseph]
vnd.groove-tool-message	[Joseph]
vnd.groove-tool-template	[Joseph]
vnd.groove-vcard	[Joseph]
vnd.ctc-posml	[Kohlhepp]
vnd.canon-lips	[Muto]
vnd.canon-cpdl	[Muto]
vnd.trueapp	[Hepler]
vnd.s3sms	[Tarkkala]
iotp	[RFC2935]
vnd.mcd	[Gotoh]
vnd.httpphone	[Lefevre]
vnd.informix-visionary	[Gales]
vnd.msign	[Borcherding]
vnd.ms-lrm	[Ledoux]
vnd.contact.cmsg	[Patz]
vnd.epson.esf	[Hoshina]
whoispp-query	[RFC2957]
whoispp-response	[RFC2958]
vnd.mozilla.xul+xml	[McDaniel]
parityfec	[RFC3009]
vnd.palm	[Peacock]
vnd.fsc.weblaunch	[D.Smith]
vnd.tve-trigger	[Welsh]
dvcs	[RFC3029]
sieve	[RFC3028]
vnd.vividence.scriptfile	[Risher]
vnd.hhe.lesson-player	[Jones]
beep+xml	[RFC3080]
font-tdpfr	[RFC3073]
vnd.mseq	[Le Bodic]
vnd.aether.imp	[Moskowitz]
vnd.Mobius.MQY	[Devasia]
vnd.Mobius.MBK	[Devasia]
vnd.vidsoft.vidconference	[Hess]
vnd.ibm.afplinedata	[Buis]

<u>Type</u>	<u>Subtype</u>	<u>Description</u>	<u>Reference</u>
<b>image</b>	jpeg		[RFC2045,RFC2046]
	gif		[RFC2045,RFC2046]
	ief	Image Exchange Format	[RFC1314]
	g3fax		[RFC1494]
	tiff	Tag Image File Format	[RFC2302]
	cgm	Computer Graphics Metafile	[Francis]
	naplps		[Ferber]
	vnd.dwg		[Moline]
	vnd.svf		[Moline]
	vnd.dxf		[Moline]
	png		[Randers-Pehrson]
	vnd.fpx		[Spencer]
	vnd.net-fpx		[Spencer]
	vnd.xiff		[SMartin]
	prs.btif		[Simon]
	vnd.fastbidsheet		[Becker]
	vnd.wap.wbmp		[Stark]
	prs.pti		[Laun]
	vnd.cns.inf2		[McLaughlin]
	vnd.mix		[Reddy]
	vnd.fujixerox.edmics-rlc		[Onda]
	vnd.fujixerox.edmics-mmr		[Onda]
	vnd.fst		[Fuldseth]
<b>audio</b>	basic		[RFC2045,RFC2046]
	32kadpcm		[RFC2421,RFC2422]
	vnd.qcelp		[Lundblade]
	vnd.digital-winds		[Strazds]
	vnd.lucent.voice		[Vaudreuil]
	vnd.octel.sbc		[Vaudreuil]
	vnd.rhetorex.32kadpcm		[Vaudreuil]
	vnd.vmx.cvsd		[Vaudreuil]
	vnd.nortel.vbk		[Parsons]
	vnd.cns.anpl		[McLaughlin]
	vnd.cns.inf1		[McLaughlin]
	L16		[RFC2586]
	vnd.everad.plj		[Cicelsky]
	telephone-event		[RFC2833]
	tone		[RFC2833]
	prs.sid		[Walleij]
	vnd.nuera.ecelp4800		[Fox]
	vnd.nuera.ecelp7470		[Fox]
	mpeg		[RFC3003]
	parityfec		[RFC3009]
	MP4A-LATM		[RFC3016]
vnd.nuera.ecelp9600		[Fox]	
G.722.1		[RFC3047]	

<u>Type</u>	<u>Subtype</u>	<u>Description</u>	<u>Reference</u>
<b>video</b>	mpeg		[RFC2045,RFC2046]
	quicktime		[Paul Lindner]
	vnd.vivo		[Wolfe]
	vnd.motorola.video		[McGinty]
	vnd.motorola.videop		[McGinty]
	vnd.fvt		[Fuldseth]
	pointer		[RFC2862]
	parityfec		[RFC3009]
	vnd.mpegurl		[Recktenwald]
	MP4V-ES		[RFC3016]
	vnd.nokia.interleaved-multimedia		[Kangaslampi]
<b>model</b>			[RFC2077]
	iges		[Parks]
	vrml		[RFC2077]
	mesh		[RFC2077]
	vnd.dwf		[Pratt]
	vnd.gtw		[Ozaki]
	vnd.flatland.3dml		[Powers]
	vnd.vtu		[Rabinovitch]
	vnd.mts		[Rabinovitch]
	vnd.gdl		[Babits]
	vnd.gs-gdl		[Babits]
	vnd.parasolid.transmit.text		[Dearnaley,Juckes]
	vnd.parasolid.transmit.binary		[Dearnaley,Juckes]

The "media-types" directory contains a subdirectory for each content type and each of those directories contains a file for each content subtype.

```

| -application-
| -audio-----
| -image-----
| -media-types-| -message-----
|               | -model-----
|               | -multipart---
|               | -text-----
|               | -video-----

```

URL = <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types>

### Character Sets

All of the character sets listed the section on Character Sets are registered for use with MIME as MIME Character Sets. The correspondance between the few character sets listed in the MIME specifications [RFC2045,RFC2046] and the list in that section are:

<u>Type</u>	<u>Description</u>	<u>Reference</u>
US-ASCII	see ANSI_X3.4-1968 below	[RFC2045,RFC2046]
ISO-8859-1	see ISO_8859-1:1987 below	[RFC2045,RFC2046]
ISO-8859-2	see ISO_8859-2:1987 below	[RFC2045,RFC2046]
ISO-8859-3	see ISO_8859-3:1988 below	[RFC2045,RFC2046]
ISO-8859-4	see ISO_8859-4:1988 below	[RFC2045,RFC2046]
ISO-8859-5	see ISO_8859-5:1988 below	[RFC2045,RFC2046]
ISO-8859-6	see ISO_8859-6:1987 below	[RFC2045,RFC2046]
ISO-8859-7	see ISO_8859-7:1987 below	[RFC2045,RFC2046]
ISO-8859-8	see ISO_8859-8:1988 below	[RFC2045,RFC2046]
ISO-8859-9	see ISO_8859-9:1989 below	[RFC2045,RFC2046]

### Access Types

<u>Type</u>	<u>Description</u>	<u>Reference</u>
FTP		[RFC2045,RFC2046]
ANON-FTP		[RFC2045,RFC2046]
TFTP		[RFC2045,RFC2046]
AFS		[RFC2045,RFC2046]
LOCAL-FILE		[RFC2045,RFC2046]
MAIL-SERVER		[RFC2045,RFC2046]
content-id		[RFC1873]

### Conversion Values

Conversion values or Content Transfer Encodings.

<u>Type</u>	<u>Description</u>	<u>Reference</u>
7BIT		[RFC2045,RFC2046]
8BIT		[RFC2045,RFC2046]
BASE64		[RFC2045,RFC2046]
BINARY		[RFC2045,RFC2046]
QUOTED-PRINTABLE		[RFC2045,RFC2046]

**MIME / X.400 MAPPING TABLES**

## MIME to X.400 Table

<b>MIME content-type</b>	<b>X.400 Body Part</b>	<b>Reference</b>
text/plain		
charset=us-ascii	ia5-text	[RFC1494]
charset=iso-8859-x	EBP - GeneralText	[RFC1494]
text/richtext	no mapping defined	[RFC1494]
application/oda	EBP - ODA	[RFC1494]
application/octet-stream	bilaterally-defined	[RFC1494]
application/postscript	EBP - mime-postscript-body	[RFC1494]
image/g3fax	g3-facsimile	[RFC1494]
image/jpeg	EBP - mime-jpeg-body	[RFC1494]
image/gif	EBP - mime-gif-body	[RFC1494]
audio/basic	no mapping defined	[RFC1494]
video/mpeg	no mapping defined	[RFC1494]

Abbreviation: EBP - Extended Body Part

X.400 to MIME Table  
Basic Body Parts

<b>X.400 Basic Body Part</b>	<b>MIME content-type</b>	<b>Reference</b>
ia5-text	text/plain; charset=us-ascii	[RFC1494]
voice	No Mapping Defined	[RFC1494]
g3-facsimile	image/g3fax	[RFC1494]
g4-class1	no mapping defined	[RFC1494]
teletex	no mapping defined	[RFC1494]
videotex	no mapping defined	[RFC1494]
encrypted	no mapping defined	[RFC1494]
bilaterally-defined	application/octet-stream	[RFC1494]
nationally-defined	no mapping defined	[RFC1494]
externally-defined	See Extended Body Parts	[RFC1494]

<b>X.400 Extended Body Part</b>	<b>MIME content-type</b>	<b>Reference</b>
GeneralText	text/plain; charset=iso-8859-x	[RFC1494]
ODA	application/oda	[RFC1494]
mime-postscript-body	application/postscript	[RFC1494]
mime-jpeg-body	image/jpeg	[RFC1494]
mime-gif-body	image/gif	[RFC1494]

**REFERENCES**

- [MacMime] Work in Progress.
- [RFC1036] Horton, M., and R. Adams, "Standard for Interchange of USENET Messages", RFC 1036, AT&T Bell Laboratories, Center for Seismic Studies, December 1987.
- [RFC1494] Alvestrand, H., and S. Thompson, "Equivalences between 1988 X.400 and RFC-822 Message Bodies", RFC 1494, SINTEF DELAB, Soft\*Switch, Inc., August 1993.
- [RFC1563] Borenstien, N., "The text/enriched MIME content-type". RFC 1563, Bellcore, January 1994.
- [RFC1767] Crocker, D., "MIME Encapsulation of EDI Objects". RFC 1767, Brandenburg Consulting, March 1995.
- [RFC1866] Berners-Lee, T., and D. Connolly, "Hypertext Markup Language - 2.0", RFC 1866, MIT/W3C, November 1995.
- [RFC1873] Levinson, E., "Message/External-Body Content-ID Access Type", RFC 1873, Accurate Information Systems, Inc. December 1995.
- [RFC1874] Levinson, E., "SGML Media Types", RFC 1874, Accurate Information Systems, Inc. December 1995.
- [RFC1892] Vaudreuil, G., "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", RFC 1892, Octel Network Services, January 1996.
- [RFC1894] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 1894, University of Tennessee, Octel Network Services, January 1996.
- [RFC1895] Levinson, E., "The Application/CALS-1840 Content Type", RFC 1895, Accurate Information Systems, February 1996.
- [RFC1896] Resnick, P., and A. Walker, "The Text/Enriched MIME Content Type", RFC 1896, Qualcomm, Intercon, February 1996.
- [RFC1945] Berners-Lee, Y., R. Feilding, and H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945. MIT/LCS, UC Irvine, MIT/LCS, May 1996.
- [RFC2045] Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [RFC2046] Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.
- [RFC2077] Nelson, S., C. Parks, and Mitra, "The Model Primary Content

Type for Multipurpose Internet Mail Extensions", RFC 2077, LLNL, NIST, WorldMaker, January 1997.

- [RFC2122] Mavrikakis, D., Layec, H., and K. Kartmann, "VEMMI URL Specification", RFC 2122, Monaco Telematique MC-TEL, ETSI, Telecommunication+Multimedia, March 1997.
- [RFC2220] Guenther, R., "The Application/MARC Content-type", RFC 2220, Library of Congress, Network Devt. & MARC Standards Office, October 1997.
- [RFC2298] Fajman, R., "An Extensible Message Format for Message Disposition Notifications", RFC 2298, March 1998.
- [RFC2302] Parsons, G., et. al., "Tag Image File Format (TIFF) - image/tiff", RFC 2302, March 1998.
- [RFC2311] Dusse, S., et. al., "S/MIME Version 2 Message Specification, RFC 2311, March 1998.
- [RFC2318] Lie, H., Bos, B., and C. Lilley, "The text/css Media Type", RFC 2318, March 1998.
- [RFC2327] Handley, M., and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1999.
- [RFC2376] Whitehead, E., and M. Murata, "XML Media Types", July 1998.
- [RFC2387] Levinson, E., "The MIME Multipart/Related Content-type", RFC 2387, Xison Inc, August 1998.
- [RFC2388] Masinter, L., "Form-based File Upload in HTML", RFC 2388, Xerox Corporation, August 1998.
- [RFC2421] Vaudreuil, G., and G. Parsons, "Voice Profile for Internet Mail - version 2", RFC 2421, September 1998.
- [RFC2422] Vaudreuil, G., and G. Parsons, "Toll Quality Voice - 32 kbit/s ADPCM MIME Sub-type Registration", RFC 2422, September 1998.
- [RFC2423] Vaudreuil, G., and G. Parsons, "VPIM Voice Message MIME Sub-type Registration", RFC 2423, September 1998.
- [RFC2425] Howes, T., Smith, M., and F. Dawson, "A MIME Content-Type for Directory Information", RFC 2425, September 1998.
- [RFC2442] Freed, N., Newman, D., Belissent, J. and M. Hoy, "The Batch SMTP Media Type", RFC 2442, November 1998.
- [RFC2445] Dawson, F., and D. Stenerson, "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998.
- [RFC2483] M. Mealling and R. Daniel, "URI resolution services necessary for URN resolution", RFC 2483, January 1999.

- [RFC2510] Adams, C., and S. Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols", RFC 2510, March 1999.
- [RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.
- [RFC2585] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP", RFC 2585, May 1999.
- [RFC2586] Salsman, J and H. Alvestrand, "The Audio/L16 MIME content type", RFC 2586, May 1999.
- [RFC2616] Fielding, R., et. al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC2652] Allen, J., and M. Mealling, "MIME Object Definitions for the Common Indexing Protocol (CIP)", RFC 2652, August 1999.
- [RFC2793] Hellstrom, G., "RTP Payload for Text Conversation", RFC 2793, May 2000.
- [RFC2833] Schulzrinne, H., "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 2833, May 2000.
- [RFC2854] Connolly, D., and L. Masinter, "The 'text/html' Media Type", RFC 2854, June 2000.
- [RFC2862] Civanlar, M., and G. Cash, "RTP Payload Format for Real-Time Pointers", RFC 2862, June 2000.
- [RFC2910] Herriot, R., Editor, Butler, S., Moore, P., Turner, R. and J. Wenn, "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2910, September 2000.
- [RFC2935] Eastlake, D. and C. Smith, "Internet Open Trading Protocol (IOTP) HTTP Supplement", RFC 2935, September 2000.
- [RFC3003] M. Nilsson, "The audio/mpeg Media Type", RFC 3003, November 2000.
- [RFC3009] J.Rosenberg and H.Schulzrinne, "Registration of parityfec MIME types", RFC 3009, November 2000.
- [RFC3016] Kikuchi, Y., T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata, "RTP payload format for MPEG-4 Audio/Visual streams", RFC 3016, November 2000.
- [RFC3028] T. Showalter, "Sieve: A Mail Filtering Language" RFC 3028, January 2001.
- [RFC3029] Adams, C., P. Sylvester, M. Zolotarev, and R. Zuccherato, "Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols", RFC 3029, January 2001.
- [RFC3047] Luthi, P. "RTP Payload Format for ITU-T Recommendation G.772.1", RFC 3047, January 2001.
- [RFC3073] Collins, J., "Portable Font Resource (PFR) - application/font-tdpfr MIME Sub-type Registration", RFC 3073, February 2001.
- [RFC3080] Rose, M., "The Blocks Extensible Exchange Protocol Core", RFC 3080, February 2001.

**PEOPLE**

[Adams] Greg Adams , March 1997.  
[Allen] Steve Allen , September 1997.  
[Alvestrand] Harald T. Alvestrand , January 1997.  
[Applebaum] David Applebaum, , February 1997.  
[Aubrey] Steve Aubrey, , July 1998.  
[Babits] Attila Babits, , April 2000, May 2000.  
[Bartram] Chris Bartram, , May 1998.  
[Becker] Scott Becker, , April 1996, October 1998.  
[Ben-Kiki] Oren Ben-Kiki, , October 1998.  
[Berners-Lee] Tim Berners-Lee, , May 1996.  
[Borcherding] Malte Borcherding, , August 2000.  
[Borenstein] Nathaniel Borenstein, , April 1994.  
[Bradley] Dan Bradley, , October 2000.  
[Buettgenbach] Gert Buettgenbach, , May 1997.  
[Buis] Roger Buis, , March 2001.  
[Butler] Tim Butler, , April 1996.  
[Larry Campbell]  
[Chandhok] Ravinder Chandhok, , December 1998.  
[Chase] Brad Chase, , May 1996.  
[Cicelsky] Shay Cicelsky, , May 2000.  
[Cole] Pete Cole, , June 1996.  
[Dave Crocker] Dave Crocker  
[Terry Crowley]  
[Daniel] Ron Daniel, Jr. , June 1997.  
[Dearnaley] Roger Dearnaley, , October 2000.  
[Dellutri] Steve Dellutri, , March 1998.  
[Devasia] Alex Devasia, , March 2001.  
[Doggett] Jay Doggett, , February 1997.  
[Domino] Michael Domino, , February 1997.  
[Duffy] Michael Duffy, , September 1997.  
[Eastlake] Donald E. Eastlake 3rd, , April 1995, May 2000.  
[Faltstrom] Patrik Faltstrom  
[Fleischman] Eric Fleischman , April 1997.  
[Floersch] Dick Floersch , March 1997.  
[Flurry] Henry Flurry , April 1999.  
[Fox] Michael Fox, , August 2000, January 2001.  
[Francis] Alan Francis, A.H.Francis@open.ac.uk, December 1995.  
[Fujii] Kiyofusa Fujii , February 1997.  
[Fuldseth] Arild Fuldseth, , June 2000.  
[Gales] Christopher Gales, , August 2000.  
[Gandert] April Gandert , April 1999.  
[Gill] Sukvinder S. Gill, , April 1996.  
[Glazer] David Glazer, , April 1995.  
[Gotoh] Tadashi Gotoh, , February 2000, June 2000.  
[Gu] Yu Gu, , December 1998.  
[Gupta] Alope Gupta , April 1999.  
[Gurak] Tom Gurak, , March 1997.  
[Gurney] John-Mark Gurney , August 1999.  
[Guy] David Guy, , June 1997.  
[Helmee] Nor Helmee, , November 1998.  
[Hepler] J. Scott Hepler, , May 2000.  
[Herzberg] Amir Herzberg, , February 1997.  
[Hess] Robert Hess, , March 2001.  
[Hodge] Tim Hodge, , August 2000.  
[Hohensee] Reinhard Hohensee , September 1997.  
[Holstage] Mary Holstage , May 1998.  
[Hoshina] Shoji Hoshina , January 1999, September 2000.  
[Hurtta] Kari E. Hurtta  
[Imoucha] Philippe Imoucha , October 1996.  
[IPTC] International Press Telecommunications Council (David Allen), July 2000.

[Jones] Randy Jones , , January 2001.  
[Joseph] Todd Joseph , February 2000, March 2000, April 2000.  
[Juckles] John Juckles , , October 2000.  
[Kangaslampi] Petteri Kangaslampi, March 2001.  
[Katz] Steve Katz , , June 1995.  
[Klos] Steven Klos , , February 1997.  
[Kohlhepp] Bayard Kohlhepp , , April 2000.  
[Korver] Brian Korver , , October 1996.  
[Laramie] Michael Laramie , February 1999.  
[Laun] Juern Laun , April 1999.  
[Le Bodic] Gwenael Le Bodic , March 2001.  
[Ledoux] Eric Ledoux , , August 2000.  
[Lefevre] Franck Lefevre , , August 2000.  
[Leow] Steve Leow , April 1999.  
[Levitt] Glenn Levitt , October 1996.  
[Lines] John Lines , January 1998.  
[Lubin] Dovid Lubin , October 1997.  
[Lubos] Mikusiak Lubos , October 1996.  
[Lundblade] Laurence Lundblade , October 1996.  
[Manning] Dave Manning , January, March 1999.  
[Martin] Bruce Martin , November 1998.  
[Martin] Steven Martin , October 1997.  
[Matsumoto] Takashi Matsumoto , February 2000  
[Matthewman] David Matthewman , October 1996.  
[McDaniel] Braden N. McDaniel , , October 2000.  
[McGinty] Tom McGinty  
[McLaughlin] Ann McLaughlin , April 1999.  
[Minnis] James Minnis , February 2000  
[Moline] Jodi Moline , , April 1996.  
[Mori] Hiroyoshi Mori , , August 1998.  
[Moskowitz] Jay Moskowitz , , March 2001.  
[Muto] Shin Muto , , May 2000.  
[Mutz] Andy Mutz , , December 1997.  
[Nagatomo] Yasuhito Nagatomo , January 1998.  
[Natarajan] Satish Natarajan , , August 1998.  
[Nelson] Jan Nelson , , May 2000.  
[Nilsson] Martin Nilsson , , October 2000.  
[O'Brien] Michael O'Brien , January 1998.  
[Ogita] Masumi Ogita , , October 1997.  
[Okudaira] Seiji Okudaira , October 1997.  
[Olsson] Thomas Olsson , April 1998.  
[Onda] Masanori Onda , February 2000.  
[Ozaki] Yutaka Ozaki , January 1999.  
[Paul Lindner]  
[Parker] David Parker , August 1999.  
[Parks] Curtis Parks , , April 1995.  
[Parsons] Glenn Parsons , February 1999.  
[Patton] Mark Patton , March 1999.  
[Patz] Frank Patz , , September 2000.  
[Peacock] Gavin Peacock , , November 2000.  
[Pentecost] Bob Pentecost , , March 1997.  
[Pharr] Paul C. Pharr , February 2000.  
[Powers] Michael Powers , , January 1998., January 1999.  
[Pratt] Jason Pratt , , August 1997.  
[Pruulmann] Jann Pruulmann , , December 1998.  
[Rabinovitch] Boris Rabinovitch , February 2000.  
[Randers-Pehrson] Glenn Randers-Pehrson , October 1996.  
[Recktenwald] Heiko Recktenwald , , November 2000.  
[Reddy] Saveen Reddy , July 1999.  
[Rehem] Yaser Rehem , , February 1997.  
[Resnick] Pete Resnick , , February 2000.  
[Risher] Mark Risher , , December 2000.  
[Rose] Marshall Rose , , April 1995.

[Rosenberg] Jonathan Rosenberg, , October 2000.  
[Rungchavalnont] Khemchart Rungchavalnont, July 1997.  
[Sandal] Troy Sandal , November 1997.  
[Santinelli] Paul Santinelli, Jr. , October 1996.  
[Scrathcley] Greg Scratchley , October 1998.  
[Searcy] Meredith Searcy, , June 1997.  
[Shapiro] Ehud Shapiro  
[Shilts] Reed Shilts , February 1999, August 1999.  
[Simon] Ben Simon, , September 1998.  
[Simonoff] Steven Simonoff , August 1999.  
[Simpson] Ray Simpson , January 1998.  
[Slawson] Dean Slawson, , May 1996.  
[Slusanschi] Horia Cristian Slusanschi , January 1998.  
[D.Smith] Derek Smith, , November 2000.  
[Smith] Nick Smith, , June 1995.  
[Smolgovsky] Roman Smolgovsky , April 1999.  
[Solomon] Monty Solomon, , February 1997.  
[Spencer] Marc Douglas Spencer , October 1996.  
[Henry Spencer]  
[Stark] Peter Stark , March 1999.  
[Strazds] Armands Strazds , January 1999.  
[Sugimoto] Masahiko Sugimoto , October 1997.  
[T.Sugimoto] Taisuke Sugimoto April 1999.  
[Sweet] Michael Sweet , July 1999.  
[Swenson] Janine Swenson , January 1998.  
[Szekely] Etay Szekely , October 1996.  
[Taguchi] Yasuo Taguchi , July 1998.  
[Tanaka] Manabu Tanaka , September 1997.  
[Tarkkala] Lauri Tarkkala, , May 2000.  
[Tidwell] Paul Tidwell , February 2000.  
[Togashi] Nobukazu Togashi , June 1997.  
[Tomasello] Luke Tomasello  
[Tupper] Jeff Tupper , February 2000.  
[Vaudreuil] Greg Vaudreuil , January 1999.  
[Walleij] Linus Walleij, , July 2000.  
[WAP-Forum] WAP Forum Ltd. , February 2000.  
[Wattenberger] Paul Wattenberger , June 1997.  
[Webb] Steve Webb , October 1996.  
[Wedel] Eric Wedel , October 1996.  
[Welsh] Linda Welsh, , November 2000.  
[Wexler] Mike Wexler, , April 1996.  
[Widener] Glenn Widener , June 1997.  
[Wohler] Bill Wohler, , July 1997.  
[Wolfe] John Wolfe, , April 1996.  
[Van Nostern] Gene C. Van Nostern , February 1995.  
[Yellow] Mr. Yellow , March 1998.  
[Yoshitake] Jun Yoshitake, , February 1997.  
[Zilles] Steve Zilles , October 1996.