Media Types

Web Architecture and Information Management [./] Spring 2009 — INFO 190-02 (CCN 42509)

Erik Wilde, UC Berkeley School of Information

2009-03-11

Contents

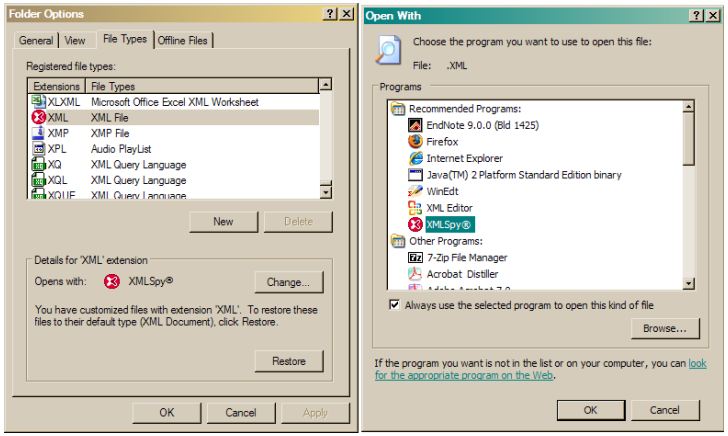
Abstract (2)

One of the most important aspect of computer-based communications is the concept of *media types*, the question what type of information some digital artifact represents, and how it is encoded. The most common standard for this information is the scheme introduced by *Multipurpose Internet Mail Extensions (MIME)*. Media types can be negotiated by peers communicating through HTTP. Some media types allow fragment identifiers, which allow references to a resource to identify a fragment of the complete resource.

Multipurpose Internet Mail Extensions (MIME) (3)

- Basic e-mail only supports ASCII text messages
- MIME was introduced in 1993 to standardize a more powerful message format
  - multiple objects in a single message
  - text having unlimited line length or overall length
  - character sets other than ASCII, allowing non-English language messages
  - binary or application specific files
  - images, audio, video and multi-media messages
- Resource types are necessary for every automated action with resources
  - Unix started with `/etc/mime.types`, a list of mappings between extensions and media types
  - the Unix `file` command uses simple fingerprints (specified in `/etc/magic`)
  - double-clicking in GUIs needs a file association (based on the file's type) to work

Windows File Type Handling (4)

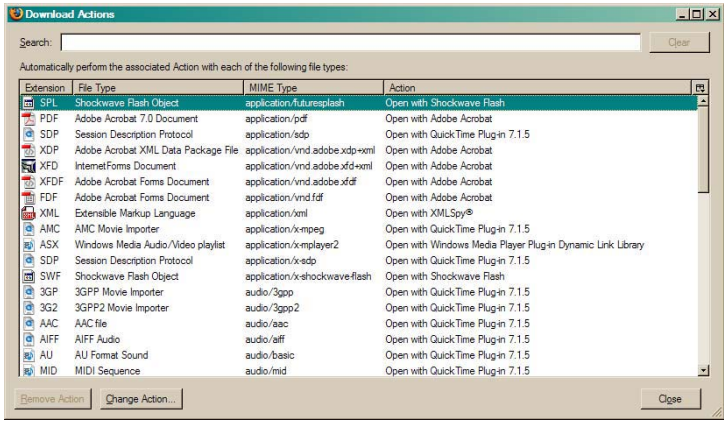


Media Types and the Web

Browsers and Resources (6)

- Web browsers retrieve resources and render them
  - HTTP can transfer any kind of resource (binary resources must be transfer encoded)
  - resource types cannot (and should not) be inferred from the URI
- HTTP combines data transfer, transfer management, and metadata
  - basic information about a resource (modification date)
  - information describing the resource's type (media type) and content (language)
- The resource type received may or may not be supported by the browser
  - *built-in support* is provided for the core Web resource types (HTML, GIF, JPEG)
  - *plug-in* support is an add-on to the browser for popular types (PDF, Flash)
  - *external applications* are standalone applications invoked by the browser

Firefox Media Type Handling (7)



Media Types

Content Types (9)

- MIME splits the word of resource types into *Content Types* and *Subtypes*
  - *Content types* are the main classification of a resource type
  - Subtypes [Subtypes (1)] qualify the format and encoding used for the content
- Content types classify the world of resource type into 8 areas
  - audio for media types representing exclusively audio signals
  - image for any media type representing two-dimensional images
  - message for resources representing e-mail messages
  - model for complex representations of physical objects (very unpopular)
  - multipart for MIME entities containing multiple individual MIME-tagged resources
  - text for mainly textual material (e.g., HTML is considered to be text)
  - video for media types combining moving pictures with audio
  - application for any resource which cannot by classified anywhere else
  - (example is only used for media type examples, not for real-world resources)

Subtypes (10)

- Within each content type, many different data formats are in use
  - content types only allow a broad classification
  - subtypes allow the identification of a specific data format of a resource
- Subtypes are expected to be registered [Media Type Registration (1)] with the IANA [http://www.iana.org/]
  - unregistered subtypes can be used but must have a x- prefix
- Additional qualifiers can be used to be more specific
  - text/plain is the media type for plain text files
  - plain text files have additional properties such as character encoding and language
  - text/plain can be further qualified to text/plain; charset=iso-8859-1

Media Type Registration (11)

- Media types need to be registered together with a documentation
  - this makes sense if it is assumed that registered types should be openly accessible
  - this becomes complicated if the types are proprietary and not publicly documented
- It makes sense to register types even if they are not publicly documented
  - if a Word document is sent by e-mail it should be opened by the Word application
  - IANA registers "vnd." prefixed subtypes with less requirements than "regular" types
  - vendor specific types are often undocumented and may change significantly over time
- Using well-defined types makes handling resources more stable
  - the IANA registry contains hundreds of types (most of them application types)
  - when designing applications dealing with various content types, use media types as the foundation

application/msword Media Type (12)

```
SECURITY CONSIDERATIONS:
None known.


PUBLISHED SPECIFICATION:

Specification by example:

   From any microsoft word application select "Save As..." from the
   "File" menu.  Enter a filename, make sure that "Normal" is specified
   for the file type, and click "Save".

Company Contact:

   Microsoft Inc.

   16011 NE 36th Way
   Box 97017
   Redmond WA, 98073-9717
```


Text Content Types

Plain Text (14)

- RFC 2046 [http://dret.net/rfc-index/reference/RFC2046] defines plain text files as a basic media type
  - any text file that does not contains structures which are intended for machine-based processing
  - even Comma-Separated Values (CSV) [Comma-Separated Values (CSV) (1)] does not count as plain text
- Guessing of character encoding is hard and unreliable and should be avoided
  - the character encoding can be specified with an additional parameter: text/plain; charset=iso-8859-1
  - if no such parameter is present, ASCII should be assumed as the character encoding
- For more specific text subtypes, various other subtypes exist [http://www.iana.org/assignments/media-types/text/]
  - calendar for information about calendar entries
  - javascript for JavaScript code (should now be marked as application/javascript)
  - sgml and xml for text with additional markup

HTML (15)

- RFC 2854 [http://dret.net/rfc-index/reference/RFC2854] registers text/html for HTML documents
  - like Plain Text [Plain Text (1)] the character encoding can also be specified as a parameter
  - it is not specific for some version of HTML (version information can be found in the HTML document)
- HTML Fragment Identifiers [HTML Fragment Identifiers (1)] are also defined by the media type registration
- HTML in many cases needs additional resources to be "self-contained"
  - images which are references by img elements (maybe external image maps)
  - other media referenced by object or applet (or the deprecated embed)
  - stylesheets or scripts which are referenced in the document head (they may reference other files …)

Comma-Separated Values (CSV) (16)

- RFC 4180 [http://dret.net/rfc-index/reference/RFC4180] defines a textual format for "spreadsheet data"
- CSV has been used for a long time, but some of the details were solved differently
- Defining a media type makes it easier for implementations to know what to expect
  - the registration not only registers the type, but also defines it
- CSV is not overly complex, but some issues have to be solved
  - how to separate lines (CRLF)
  - how to end the file (CRLF is allowed but optional)
  - are there headers allowed (yes, but they are not marked as such)
  - may different lines use different numbers of fields (no)
  - are spaces significant (yes)
  - are quotes significant (no, they are delimiters, so quotes as values must be escaped)
  - how to treat fields with CRLF, commas, or quotes (enclose the value in quotes)

Image Content Types

Graphic Interchange Format (GIF) (18)

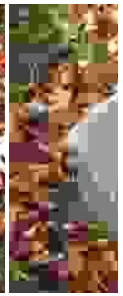
- RFC 2046 [http://dret.net/rfc-index/reference/RFC2046] registers the oldest graphics format on the Web
- GIF was subject of a long patent debate
  - the compression technique of GIF (LZW [http://en.wikipedia.org/wiki/Lzw]) had been patented by Unisys (1983)
  - Unisys wanted to get licensing fees from all commercial online uses of GIF
  - Portable Network Graphics (PNG) [Multimedia Content; Portable Network Graphics (PNG) (1)] was developed as an effort to develop a copyright-free format
  - in 1999, Unisys changed its tactics and wanted to collect one-time fees ($5000-$7500) from all users
  - all GIF-related LZW expired in 2003/2004, so GIF is freely available now
- GIF's poor features make PNG the better choice anyway
  - 8 bit color (requires dithering for photographs), binary transparency
  - GIF's animation feature is the only thing that is not available in PNG …

Joint Photographic Experts Group (JPEG) (19)

- RFC 2046 [http://dret.net/rfc-index/reference/RFC2046] standardizes the second popular image format for the Web
- JPEG has been specifically designed for photographs
  - it always is lossy (it cannot preserve the complete information from a random bitmap)
  - it uses perception-based compression (for example, color precision is sacrificed for brightness)

Q = 50, filesize 15,138 bytes

Q = 10, filesize 4,787 bytes



Q =

Portable Network Graphics (PNG) (20)

- PNG is registered as image/png and is the third major image format
  - PNG was intended to be a royalty- and copyright-free replacement of GIF [Multimedia Content; Graphics Interchange Format (GIF) (1)]
  - image formats need to supported by browsers and thus take a long time until they are established
  - IE6 implements PNG in a very rudimentary form, IE7 handles PNG correctly
- PNG has some advantages over GIF and JPEG
  - lossless, compressed palette, grayscale, or true color images
  - 8 bit alpha channel for gradual opacity (blending into the background)
- JPEG still is the preferred format for photographic pictures
- GIF still is the preferred format for animated images

Fragment Identifiers

Identification of Resource Fragments (22)

- URIs identify a resource (based on a scheme and a scheme-specific part)
  - URIs do not necessarily identify a specific representation of a resource
  - any representation-specific operation needs to look at the resource type
- Fragment identifiers can be used to identify a part of a resource
  - http://dret.net/lectures/web-spring09/mime#frag-id [http://dret.net/lectures/web-spring09/mime#frag-id]
  - fragments are a *client side* concept (the HTTP GET requests the complete resource)
  - if the client supports fragment handling, the identifier is interpreted

HTML Fragment Identifiers (23)

- HTML allows to address named/identified elements in the HTML document
  - the first HTML versions required named `<a name="html-frag-id">incoming anchors</a>`
  - newer HTML versions allow `<div id="html-frag-id">every element to have an id</div>` [http://dret.net/lectures/web-spring09/mime#html-frag-id]
  - browsers support both ways, but the id variant should be preferred
- Only named/identified fragments can be identified
  - 99.99% of all page authors do not routinely add identifications
  - tools may be smarter and take over that task (e.g., *Movable Type* identifies all relevant elements)
  - for most pages on the Web this means users cannot link to most elements in them
- Keeping fragment identifiers stable should be a goal of Web authors
  - identify the key fragments (and maybe provide a better UI than "view source")
  - never change identifiers once they have been assigned

Conclusions (24)

- Handling Web resources and technologies requires a common vocabulary
- Media types are a useful vocabulary for identifying resource types
- Fragment identifiers add some complexity, in particular for resource variants