Internet Engineering Task Force	L. Masinter
Internet-Draft	Adobe
Intended status: Informational	September 23, 2010
Expires: March 27, 2011	

TOC

Internet Media Types and the Web draft-masinter-mime-web-info-00

Abstract

This document describes some of the ways in which parts of the MIME system, originally designed for electronic mail, have been used in the web, and some of the ways in which those uses have resulted in difficulties. This informational document is intended as background and justification for a companion Best Current Practice which makes some changes to the registry of Internet Media Types and other specifications and practices, in order to facilitate Web application design and standardization.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”

This Internet-Draft will expire on March 27, 2011.

Table of Contents

TOC

1. Introduction

This document was prompted by a set of discussions in the W3C Technical Architecture Group about web architecture and the difficulties surrounding evolution of the web, Internet Media types, multiple specifications for a single media type, and related discussions. The goal of the document is to prompt an evolution within W3C and IETF over the use of MIME (and in particular Internet Media Types) to fix some of the outstanding problems. This is an initial version review and update. The goal is to initially survey the current situation and then make a set of recommendation to the definition and use MIME components (and specifically, Internet Media Types and charset declarations) to facilitate their standardization across Web and Web-related technologies with other Internet applications. Discussion of this document is suggested on the mailing list www-tag@w3c.org, a mailing list open for subscription to all, archives at http://lists.w3.org/Archives/Public/www-tag/.

TOC

2. History

TOC

2.1. Origins of MIME

MIME was invented originally for email, based on general principles of 'messaging', a foundational architecture framework. The role of MIME was to extend Internet email messaging from ASCII-only plain text, to include other character sets, images, rich documents, etc.) The basic architecture of complex content messaging is:

- Message sent from A to B.
- Message includes some data. Sender A includes standard 'headers' telling recipient B enough information that recipient B knows how sender A intends the message to be interpreted.
- Recipient B gets the message, interprets the headers for the data and uses it as information on how to interpret the data.

MIME is a "tagging and bagging" specification:

tagging
: how to label content so the intent of how the content should be interpreted is known

bagging
: how to wrap the content so the label is clear, or, if there are multiple parts to a single

message, how to combine them.

"MIME types" (renamed "Internet Media Types") were part of the tagging -- a name space for describing how to initiate interpretation of a message. The "Internet Media Type registry" (MIME type registry) is where someone can tell the world what a particular label means, as far as the sender's intent of how recipients should process a message of that type, and the description of a recipients capability and ability for senders.

TOC

2.2. Introducing MIME into the Web

The original World Wide Web (the 0.9 version of HTTP) didn't have "tagging and bagging" -- everything sent via HTTP was assumed to be HTML. However, at the time (early 1990's) other distributed information access systems, including Gopher (distributed menu system) and WAIS (remote access to document databases) were adding capabilities for accessing many things other text and hypertext and the WWW folks were considering type tagging. It was agreed that HTTP should use MIME as the vocabulary for talking about file types and character sets. The result was that HTTP 1.0 added the "content-type" header, following (more or less) MIME. Later, for content negotiation, additional uses of this technology (in 'Accept' headers) were also added.

The differences between the use of Internet Media Types between email and HTTP were minor:

- default charset
- requirement for CRLF in plain text.

These minor differences have caused a lot of trouble.

TOC

2.3. Distributed Extensibility

The real advantage of using Internet Media Types to label content meant that the web was no longer restricted to a single format. This one addition meant expanding from Global Hypertext to Global Hypermedia (as suggested in **a 1992 email** [connolly92])

> The Internet currently serves as the backbone for a global hypertext. FTP and email provided a good start, and the gopher, WWW, or WAIS clients and servers make wide area information browsing simple. These systems even interoperate, with email servers talking to FTP servers, WWW clients talking to gopher servers, on and on.
>
> This currently works quite well for text. But what should WWW clients do as Gopher and WAIS servers begin to serve up pictures, sounds, movies, spreadsheet templates, postscript files, etc.? It would be a shame for each to adopt its own multimedia typing system.
>
> If they all adopt the MIME typing system (and as many other features from MIME as are appropriate), we can step from global hypertext to global hypermedia that much easier.

The fact that HTTP could reliably transport images of different formats, for example, allowed NCSA to add <img> to HTML. MIME allowed other document formats (Word, PDF, Postscript) and other kinds of hypermedia, as well as other applications, to be part of the web. MIME was arguably the most important extensibility mechanism in the web.

TOC

3. Problems with application to the Web

Unfortunately, while the use of Internet Media Types for the web added incredible power, several problems have arisen.

TOC

3.1. Differences between email and web delivery

Some of the differences between the application contexts of email and web delivery determine different requirements:

- web "messages" are generally HTTP responses to a specific request; this means you know more about the data before you receive it. In particular, the data really does have a 'name' (mainly, the URL used to access the data), while in messaging, the messages were anonymous.
- You would like to know more about the content before you retrieve it. The "tagging" is often not sufficient to know, for example, "can I interpret this if I retrieve it", because of versioning, capabilities, or dependencies on things like screen size or interaction capabilities of the recipient.
- Some content isn't delivered over the HTTP (files on local file system), or there is no opportunity for tagging (data delivered over FTP) and in those cases, some other ways are needed for determining file type.

Operating systems use using, and continued to evolve to use, different systems to determine the 'type' of something, different from the MIME tagging and bagging:

- 'magic numbers': in many contexts, file types could be guessed pretty reliably by looking for headers.
- Originally MAC OS had a 4 character 'file type' and another 4 character 'creator code' for file types.
- Windows evolved to use the "file extension" -- 3 letters (and then more) at the end of the file name

Information about these other ways of determining type (rather than by the content-type label) were gathered for the Internet Media Type registry; those registering types are encouraged to also describe 'magic numbers', Mac file type, common file extensions. However, since there was no formal use of that information, the quality of that information in the registry is haphazard.

Finally, there was the fact that tagging and bagging might be OK for unilaterally initiated (one-way) messaging, you might want to know whether you could handle the data before reading it in and interpreting it, but the Internet Media Types weren't enough to tell.

TOC

3.2. The Rules Weren't Quite Followed

The behavior of the community when the Internet Media Type registry was designed haven't matched expectations:

- Lots of file types aren't registered (no entry in IANA for file types).
- Those that are, the registration is incomplete or incorrect (people doing registration didn't understand 'magic number' or other fields).
- The actual content deployed or created by deployed software doesn't match the registration.

In particular, web implementations of Internet Media Types diverged from expected behavior:

- Browser implementors would be liberal in what they accepted, and use file extension and/or magic number or other 'sniffing' techniques to decide file type, without assuming content-label was authoritative. This was necessary anyway for files that weren't delivered by HTTP.
- HTTP server implementors and administrators didn't supply ways of easily associating the 'intended' file type label with the file, resulting in files frequently being delivered with a label other than the one they would have chosen if they'd thought about it, and if browsers *had* assumed content-type was authoritative. Some popular servers had default configuration files that treated any unknown type as "text/plain" (plain ext in ASCII). Since it didn't matter (the browsers worked anyway), it was hard to get this fixed.

Incorrect senders coupled with liberal readers wind up feeding a negative feedback loop based on the robustness principle.

TOC

3.3. Consequences

The result, alas, is that the web is unreliable, in that

- servers sending responses to browsers don't have a good guarantee that the browser won't "sniff" the content and decide to do something other than treat it as it is labeled
- browsers receiving content don't have a good guarantee that the content isn't mis-labeled
- intermediaries (gateways, proxies, caches, and other pieces of the web infrastructure) don't have a good way of telling what the conversation means.

This ambiguity and 'sniffing' also applies to packaged content in webapps ('bagging' but using ZIP rather than MIME multipart). (NOTE: NEEDS EXPANSION)

TOC

3.4. The Down Side of Extensibility

Extensibility adds great power, and allows the web to evolve without committee approval of every extension. For some (those who want to extend and their clients who want those extensions), this is power! For others (those who are building web components or infrastructure), extensibility is a drawback -- it adds to the unreliability and difference of the web experience. When senders use extensions recipients aren't aware of, implement incorrectly or incompletely, then communication often fails. With messaging, this is a serious problem, although most 'rich text' documents are still delivered in multiple forms (using multipart/alternative).

If your job is to support users of a popular browser, however, where each user has installed a different configuration of file handlers and extensibility mechanisms, MIME may appear to add unnecessary complexity and variable experience for users of all but the most popular types.

TOC

4. Additional considerations

This section notes some additional considerations.

TOC

4.1. There are related problems with charsets

MIME includes provisions not only for file 'types', but also, importantly the "character encoding" used by text types: for example, simple US ASCII, Western European ISO-8859-1, Unicode UTF8. A similar vicious cycle also happened with character set labels: mislabeled content happily processed correctly by liberal browsers encouraged more and more sites to proliferate text with mis-labeled character sets, to the point where browsers feel they *have* to guess the wrong label. (NEEDS EXPANSION)

There are sites that intentionally label content as iso-2022-jp or euc-jp when it is in fact one of the Microsoft extension charsets (e.g., for access to circled digits. This is an intentional misuse of the definitions of the charsets themselves -- definitions which originated at the national standards body level.

TOC

4.2. Embedded, downloaded, launch independent application

The type of a document might be determined not only for entire documents "HTML" vs "Word" vs "PDF", but also to embedded components of documents, "JPEG image" vs. "PNG image". However, the use cases, requirements and likely operational impact of MIME handling is likely different for

those use cases.

TOC

4.3. Additional Use Cases: Polyglot and Multiview

There are some interesting additional use cases which add to the design requirements:

- "Polyglot" documents: A 'polyglot' document is one which is some data which can be treated as two different Internet Media Types, in the case where the meaning of the data is the same. This is part of a transition strategy to allow content providers (senders) to manage, produce, store, deliver the same data, but with two different labels, and have it work equivalently with two different kinds of receivers (one of which knows one Internet Media Type, and another which knows a second one.) This use case was part of the transition strategy from HTML to an XML-based XHTML, and also as a way of a single service offering both HTML-based and XML-based processing (e.g., same content useful for news articles and web pages.
- "Multiview" documents: This use case seems similar but it's quite different. In this case, the same data has very different meaning when served as two different content-types, but that difference is intentional; for example, the same data served as text/html is a document, and served as an RDFa type is some specific data.

TOC

4.4. Evolution, Versioning, Forking

Formats and their specifications evolve over time -- some times compatibly, some times not. It is part of the responsibility of the designer of a new version of a file type to try to insure both forward and backward compatibility: new documents work reasonably (with some fallback) with old viewers and that old documents work reasonably with new viewers. In some cases this is accomplished, others not; in some cases, "works reasonably" is softened to "either works reasonably or gives clear warning about nature of problem (version mismatch)."

In MIME, the 'tag', the Internet Media Type, corresponds to the versioned series. Internet Media Types do not identify a particular version of a file format. Rather, the general idea is that the Internet Media Type identifies the family, and also how you're supposed to otherwise find version information on a per-format basis. Many (most) file formats have an internal version indicator, with the idea that you only need a new Internet Media Type to designate a completely incompatible format. The notion of an "Internet Media Type" is very course-grained. The general approach to this has been that the actual Media Type includes provisions for version indicator(s) embedded in the content itself to determine more precisely the nature of how the data is to be interpreted. That is, the message itself contains further information.

Unfortunately, lots has gone wrong in this scenario as well -- processors ignoring version indicators encouraging content creators to not be careful to supply correct version indicators, leading to lots of content with wrong version indicators.

Those updating an existing Internet Media Type registration to account for new versions are admonished to not make previously conforming documents non-conforming. This is harder to enforce than would seem, because the previous specifications are not always accurate to what the Internet Media Type was used for in practice.

(NOTE: MULTIPLE INCOMPATIBLE AUTHORITATIVE SPECS)

TOC

4.5. Content Negotiation

The general idea of content negotiation is when party A communicates to party B, and the message can be delivered in more than one format (or version, or configuration), there can be some way of allowing some negotiation, some way for A to communication to B the available options, and for B to be able to accept or indicate preferences.

Content negotiation happens all over. When one fax machine twirps to another when initially connecting, they are negotiating resolution, compression methods and so forth. In Internet mail, which is a one-way communication, the "negotiation" consists of the sender preparing and sending multiple versions of the message, one in text/html, one in text/plain, for example, in sender-preference order. The recipient then chooses the first version it can understand.

HTTP added "Accept" and "Accept-language" to allow content negotiation in HTTP GET, based on Internet Media Types, and there are other methods explained in the HTTP spec.

TOC

4.6. Fragment identifiers

The web added the notion of being able to address part of a content and not the whole content by adding a 'fragment identifier' to the URL that addressed the data. Of course, this originally made sense for the original web with just HTML, but how would it apply to other content. The URL spec glibly noted that "the definition of the fragment identifier meaning depends on the Internet Media Type", but unfortunately, few of the Internet Media Type definitions included this information, and practices diverged greatly.

If the interpretation of fragment identifiers depends on the MIME type, though, this really crimps the style of using fragment identifiers differently if content negotiation is wanted.

TOC

5. Where we need to go

Many people are confused about the purpose of MIME in the web, its uses, the meaning of Internet Media Types. Many W3C specifications TAG findings and Internet Media Type registrations make what are (IMHO) incorrect assumptions about the meaning and purposes of a Internet Media Type registration.

We need a clear direction on how to make the web more reliable, not less. We need a realistic transition plan from the unreliable web to the more reliable one. Part of this is to encourage senders (web servers) to mean what they say, and encourage recipients (browsers) to give preference to what the senders are sending.

We should try to create specifications for protocols and best practices that will lead the web to more reliable and secure communication. To this end, we give an overall architectural approach to use of MIME, and then specific specifications, for HTTP clients and servers, Web Browsers in general, proxies and intermediaries, which encourage behavior which, on the one hand, continues to work with the already deployed infrastructure (of servers, browsers, and intermediaries), but which advice, if followed, also improves the operability, reliability and security of the web.

NOTE: This section should be elaborated to include requirements for changes to MIME and Internet Media Type registrations to improve the situation.

TOC

6. Specific recommendations

NOTE: We should try to get agreement on the background, problem statement and requirements, before sending out any more about possible solutions. The intention is that recommendations for changes to IETF-specified processes and registries would be moved into a new BCP-track document.

However, the following is a partial list of documents that should be reviewed and updated, or new documents written.

TOC

6.1. Internet Media Type registration

Update the Internet Media Type registration process (via a new IETF BCP document):

- Allow commenting or easier update; not all Internet Media Type owners need or have all the information the internet needs. Wiki for Internet Media Types as well as formal registry? Ability to add comments about deployed senders, deployed content, deployed recievers for new recievers or senders.
- Be clearer about relationship of 'magic numbers' to sniffing; review Internet Media Types already registered and update.
- Be clearer about requiring Security Considerations to address risks of sniffing
- require definition of fragment identifier applicability
- ask the 'applications that use this type' section to be clearer about whether the file type is suitable for embedding (plug-in) or as a separate document with auto-launch (MIME handler), or should always be donwloaded.
- Be clearer about file extension use and relationship of file extensions to MIME handlers

---

TOC

6.2. Sniffing

Various new specifications promote the use of 'sniffing' -- using the content of the data to supplement or even override the declared content-type or charset. Update these specifications:

- Sniffing uses MIME registry for 'magic numbers'
- all sniffing can be a priviledge upgrade, if there is a buggy recipient, although bugs can be fixed.
- discourage sniffing unless there is no type label:
    - malformed content-type: error
    - no knowledge that given content-type isn't better than guessed content-type

---

TOC

6.3. Other specifications and BCPs

- FTP specifications: do FTP clients also change rules about guessing file types based on OS of FTP server?
- update Tag finding on authoritative metadata: is it possible to remove 'authority'?
- new: MIME and Internet Media Type section to WebArch, referencing this memo
- New: Add a W3C web architecture material on MIME in HTML to W3C web site, referencing this memo
- Reconsider other extensibility mechanisms (namespaces, for example): should they use MIME or something like it?

TOC

7. Acknowledgements

This document is the result of discussions among many individuals in the IETF and W3C.

TOC

8. IANA Considerations

This memo includes no request to IANA.

TOC

9. Security Considerations

This document discusses some of the security issues resulting from use (and mis-use) of MIME content types in the web.

---

10. Informative References

TOC

[connolly92] Connolly, D., "**Global Hypermedia**," Oct 1992.

Author's Address

TOC

Larry Masinter
Adobe
345 Park Ave.
San Jose, 95110
USA
Phone: +1 408 536 3024
Email: masinter@adobe.com
URI: http://larry.masinter.net