Internet Draft: draft-dorner-content-header-01.txt
Category: Informational

Rens Troost
Steve Dorner
January 1994

Communicating Presentation Information in Internet Messages: The Content-Disposition Header

Status of this Memo

1. Abstract

This memo provides a mechanism whereby messages conforming to the [RFC 1521] ("MIME") specification can convey presentational information. It specifies a new "Content-Disposition" header, optional and valid for any [RFC 1521] entity ("message" or "body part"). Two values for this header are described in this memo; one for the ordinary linear presentation of the body part, and another to facilitate the use of mail to transfer files. It is expected that more values will be defined in the future, and procedures are defined for extending this set of values.

This document is intended as an extension to [RFC 1521]. As such, the reader is assumed to be familiar with [RFC 1521], and [RFC 822]. The information presented herein supplements but does not replace that found in those documents.

2. Introduction

[RFC 1521] describes a standard format for encapsulating multiple pieces of data into a single Internet message. That document does not address the issue of presentation styles; it provides a framework for the interchange of message content, but leaves presentation issues solely in the hands of mail user agent (MUA) implementors.

Two common ways of presenting multipart electronic messages are as a main document with a list of separate attachments, and as a single document with the various parts expanded (displayed) inline. The display of an attachment is generally construed to require positive action on the part of the recipient, while inline message components are displayed automatically when the message is viewed. A mechanism is needed to allow the sender to transmit this sort of presentational information to the recipient; the Content-Disposition header provides this mechanism, allowing each component of a message to be tagged with an indication of its desired presentation semantics.

Tagging messages in this manner will often be sufficient for basic message formatting. However, in many cases a more powerful and flexible approach will be necessary. The definition of such approaches is beyond the scope of this memo; however, such approaches can benefit from additional Content-Disposition values and parameters, to be defined at a later date.

In addition to allowing the sender to specify the presentational disposition of a message component, it is desirable to allow her to indicate a default archival disposition; a filename. The optional "filename" parameter provides for this.

3. The Content-Disposition Header Field

Content-Disposition is an optional header; in its absence, the MUA may use whatever presentation method it deems suitable.

It is desirable to keep the set of possible disposition types small and well defined, to avoid needless complexity. Even so, evolving usage will likely require the definition of additional disposition types or parameters, so the set of disposition values is extensible; see below.

In the extended BNF notation of [RFC 822], the Content-Disposition header field is defined as follows:


```
disposition := "Content-Disposition" ":"
               disposition-type
               *(";" disposition-parm)

disposition-type := "inline"
                  / "attachment"
                  / extension-token
                  ; values are not case-sensitive

disposition-parm := filename-parm / parameter

filename-parm := "filename" "=" value;
```

`Extension-token', `parameter' and `value' are defined according to [RFC 822] and [RFC 1521].

3.1 The Inline Disposition Type

A bodypart should be marked `inline' if it is intended to be displayed automatically upon display of the message. Inline bodyparts should be presented in the order in which they are encountered, subject to the normal semantics of multipart messages.

3.2 The Attachment Disposition Type

Bodyparts can be designated `attachment' to indicate that

they are separate from the main body of the mail message, and that their display should not be automatic, but contingent upon some further action of the user. The MUA might instead present the user of a bitmap terminal with an iconic representation of the attachments, or, on character terminals, with a list of attachments from which the user could select for viewing or storage.

3.3 The Filename Parameter

The sender may want to suggest a filename to be used if the entity is detached and stored in a separate file. If the receiving MUA writes the entity to a file, the suggested filename should be used as a basis for the actual filename, where possible.

It is important that the receiving MUA not blindly use the suggested filename. The suggested filename should be checked (and possibly changed) to see that it conforms to local filesystem conventions, does not overwrite an existing file, and does not present a security problem (see Security Considerations below).

The receiving MUA should not respect any directory path information that may seem to be present in the filename parameter. The filename should be treated as a terminal component only. Portable specification of directory paths might possibly be done in the future via a separate Content-Disposition parameter, but no provision is made for it in this draft.

Current [RFC 1521] grammar restricts parameter values (and hence Content-Disposition filenames) to US-ASCII. We recognize the great desirability of allowing arbitrary character sets in filenames, but it is beyond the scope of this document to define the necessary mechanisms. We expect that the basic [RFC 1521] `value' specification will someday be amended to allow use of non-US-ASCII characters, at which time the same mechanism should be used in the Content-Disposition filename parameter.

Beyond the limitation to US-ASCII, the sending MUA may wish to bear in mind the limitations of common filesystems. Many have severe length and character set restrictions. Short alphanumeric filenames are least likely to require modification by the receiving system.

The presence of the filename parameter does not force an implementation to write the entity to a separate file. It is perfectly acceptable for implementations to leave the entity as part of the normal mail stream unless the user requests otherwise. As a consequence, the parameter may be used on any MIME entity, even `inline' ones. These will not normally be written to files, but the parameter could be used to provide a filename if the receiving user should choose to write the part to a file.

3.4 Future Extensions and Unrecognized Disposition Types

In the likely event that new parameters or types are needed, they should be registered with the IANA, in the manner specified in [RFC 1521], appendix E.

Once new types and parameters are defined, there is of course the likelihood that implementations will see types and parameters they do not understand. Furthermore, since x-tokens are allowed, implementations may also see entirely unregistered types and parameters.

Unrecognized parameters should be ignored. Unrecognized types should be treated as `attachment'. The choice of `attachment' for unrecognized types is made because a sender who goes to the trouble of producing a Content-Disposition header with a new value is more likely aiming for something more elaborate

than inline presentation.

Unless noted otherwise in the definition of a parameter, Content-Disposition parameters are valid for all dispositions. (In contrast to [RFC 1521] content-type parameters, which are defined on a per-content-type basis.)

Thus, for example, the `filename' parameter still means the name of the file to which the part should be written, even if the disposition itself is unrecognized.

3.5 Content-Disposition and Multipart

If a Content-Disposition header is used on a multipart body part, it applies to the multipart as a whole, not the individual subparts. The disposition types of the subparts do not need to be consulted until the multipart itself is presented. When the multipart is displayed, then the dispositions of the subparts should be respected.

If the `inline' disposition is used, the multipart should be displayed as normal; however, an `attachment' subpart should require action from the user to display.

If the `attachment' disposition is used, presentation of the multipart should not proceed without explicit user action. Once the user has chosen to display the multipart, the individual subpart dispositions should be consulted to determine how to present the subparts.

3.6 Content-Disposition and the Main Message

It is permissible to use Content-Disposition on the main body of an [RFC 822] message.

4. Examples

Here is a an example of a body part containing a JPEG image that is intended to be viewed by the user immediately:

```
Content-Type: image/jpeg
Content-Disposition: inline
Content-Description: just a small picture of me

<jpeg data>
```

The following body part contains a JPEG image that should be displayed to the user only if the user requests it. If the JPEG is written to a file, the file should be named

"genome.jpg":

```
Content-Type: image/jpeg
Content-Disposition: attachment; filename=genome.jpeg
Content-Description: a complete map of the human genome

<jpeg data>
```

The following is an example of the use of the `attachment' disposition with a multipart body part. The user should see text-part-1 immediately, then take some action to view multipart-2. After taking action to view multipart-2, the user will see text-part-2 right away, and be required to take action to view jpeg-1. Subparts are indented for clarity; they would not be so indented in a real message.

```
Content-Type: multipart/mixed; boundary=outer
Content-Description: multipart-1

--outer
  Content-Type: text/plain
  Content-Disposition: inline
  Content-Description: text-part-1

  Some text goes here

--outer
  Content-Type: multipart/mixed; boundary=inner
  Content-Disposition: attachment
  Content-Description: multipart-2

  --inner
    Content-Type: text/plain
    Content-Disposition: inline
    Content-Description: text-part-2

    Some more text here.

  --inner
    Content-Type: image/jpeg
    Content-Disposition: attachment
    Content-Description: jpeg-1

    <jpeg data>
  --inner--
--outer--
```

5. Summary

Content-Disposition takes one of two values, `inline' and `attachment'. 'Inline' indicates that the entity should be immediately displayed to the user, whereas `attachment' means that the user should take additional action to view the entity.

The `filename' parameter can be used to suggest a filename for storing the bodypart, if the user wishes to store it in an external file.

6. Security Considerations

There are security issues involved any time users exchange data. While these are not to be minimized, neither does this memo change the status quo in that regard, except in one instance.

Since this memo provides a way for the sender to suggest a filename, a receiving MUA must take care that the sender's suggested filename does not represent a hazard. Using UNIX as an example, some hazards would be:

+ Creating startup files (e.g., ".login").

+ Creating or overwriting system files (e.g., "/etc/passwd").

+ Overwriting any existing file.

+ Placing executable files into any command search path (e.g., "~/bin/more").

+ Sending the file to a pipe (e.g., "| sh").

In general, the receiving MUA should never name or place the file such that it will get interpreted or executed without the user explicitly initiating the action.

It is very important to note that this is not an exhaustive list; it is intended as a small set of examples only. Implementors must be alert to the potential hazards on their target systems.

7. Acknowledgements

We gratefully acknowledge the help these people provided during the preparation of this draft:

Nathaniel Borenstein
Ned Freed
Keith Moore
Dave Crocker
Dan Pritchett

8. Authors' Addresses

Author: Rens Troost rens@imsi.com

Co-Author: Steve Dorner sdorner@qualcomm.com

9. References

[RFC 1521]
Borenstein N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and

Describing the Format of Internet Message Bodies",
RFC 1521, Bellcore, Innosoft, September 1993.

[RFC 822]
Crocker, D., "Standard for the Format of ARPA Internet
Text Messages", STD 11, RFC 822, UDEL, August 1982.