

The MIME Multipart/Related Content-type

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

The Multipart/Related content-type provides a common mechanism for representing objects that are aggregates of related MIME body parts. This document defines the Multipart/Related content-type and provides examples of its use.

1. Introduction

Several applications of MIME, including MIME-PEM, and MIME-Macintosh and other proposals, require multiple body parts that make sense only in the aggregate. The present approach to these compound objects has been to define specific multipart subtypes for each new object. In keeping with the MIME philosophy of having one mechanism to achieve the same goal for different purposes, this document describes a single mechanism for such aggregate or compound objects.

The Multipart/Related content-type addresses the MIME representation of compound objects. The object is categorized by a "type" parameter. Additional parameters are provided to indicate a specific starting body part or root and auxiliary information which may be required when unpacking or processing the object.

Multipart/Related MIME entities may contain Content-Disposition headers that provide suggestions for the storage and display of a body part. Multipart/Related processing takes precedence over Content-Disposition; the interaction between them is discussed in section 4.

Responsibility for the display or processing of a Multipart/Related's constituent entities rests with the application that handles the compound object.

2. Multipart/Related Registration Information

The following form is copied from RFC 1590, Appendix A.

To: IANA@isi.edu Subject: Registration of new Media Type content-type/subtype

Media Type name: Multipart

Media subtype name: Related

Required parameters: Type, a media type/subtype.

Optional parameters: Start
Start-info

Encoding considerations: Multipart content-types cannot have encodings.

Security considerations: Depends solely on the referenced type.

Published specification: RFC-REL (this document).

Person & email address to contact for further information:

Edward Levinson
47 Clive Street
Metuchen, NJ 08840-1060
+1 908 494 1606
Xison@cnj.digex.net

3. Intended usage

The Multipart/Related media type is intended for compound objects consisting of several inter-related body parts. For a Multipart/Related object, proper display cannot be achieved by individually displaying the constituent body parts. The content-type of the Multipart/Related object is specified by the type parameter. The "start" parameter, if given, points, via a content-ID, to the body part that contains the object root. The default root is the first body part within the Multipart/Related body.

The relationships among the body parts of a compound object distinguishes it from other object types. These relationships are often represented by links internal to the object's components that reference the other components. Within a single operating environment the links are often file names, such links may be represented within a MIME message using content-IDs or the value of some other "Content-" headers.

3.1. The Type Parameter

The type parameter must be specified and its value is the MIME media type of the "root" body part. It permits a MIME user agent to determine the content-type without reference to the enclosed body part. If the value of the type parameter and the root body part's content-type differ then the User Agent's behavior is undefined.

3.2. The Start Parameter

The start parameter, if given, is the content-ID of the compound object's "root". If not present the "root" is the first body part in the Multipart/Related entity. The "root" is the element the applications processes first.

3.3. The Start-Info Parameter

Additional information can be provided to an application by the start-info parameter. It contains either a string or points, via a content-ID, to another MIME entity in the message. A typical use might be to provide additional command line parameters or a MIME entity giving auxiliary information for processing the compound object.

Applications that use Multipart/Related must specify the interpretation of start-info. User Agents shall provide the parameter's value to the processing application. Processes can distinguish a start-info reference from a token or quoted-string by examining the first non-white-space character, "<" indicates a reference.

3.4. Syntax

```
related-param    := [ ";" "start" "=" cid ]
                  [ ";" "start-info" "="
                    ( cid-list / value ) ]
                  [ ";" "type" "=" type "/" subtype ]
                  ; order independent

cid-list         := cid cid-list
```

```
cid           := msg-id       ; c.f. [822]

value         := token / quoted-string ; c.f. [MIME]
               ; value cannot begin with "<"
```

Note that the parameter values will usually require quoting. Msg-id contains the special characters "<", ">", "@", and perhaps other special characters. If msg-id contains quoted-strings, those quote marks must be escaped. Similarly, the type parameter contains the special character "/".

4. Handling Content-Disposition Headers

Content-Disposition Headers [DISP] suggest presentation styles for MIME body parts. [DISP] describes two presentation styles, called the disposition type, INLINE and ATTACHMENT. These, used within a multipart entity, allow the sender to suggest presentation information. [DISP] also provides for an optional storage (file) name. Content-Disposition headers could appear in one or more body parts contained within a Multipart/Related entity.

Using Content-Disposition headers in addition to Multipart/Related provides presentation information to User Agents that do not recognize Multipart/Related. They will treat the multipart as Multipart/Mixed and they may find the Content-Disposition information useful.

With Multipart/Related however, the application processing the compound object determines the presentation style for all the contained parts. In that context the Content-Disposition header information is redundant or even misleading. Hence, User Agents that understand Multipart/Related shall ignore the disposition type within a Multipart/Related body part.

It may be possible for a User Agent capable of handling both Multipart/Related and Content-Disposition headers to provide the invoked application the Content-Disposition header's optional filename parameter to the Multipart/Related. The use of that information will depend on the specific application and should be specified when describing the handling of the corresponding compound object. Such descriptions would be appropriate in an RFC registering that object's media type.

5. Examples

5.1 Application/X-FixedRecord

The X-FixedRecord content-type consists of one or more octet-streams and a list of the lengths of each record. The root, which lists the record lengths of each record within the streams. The record length list, type Application/X-FixedRecord, consists of a set of INTEGERS in ASCII format, one per line. Each INTEGER gives the number of octets from the octet-stream body part that constitute the next "record".

The example below, uses a single data block.

```
Content-Type: Multipart/Related; boundary=example-1
      start="<950120.aaCC@Xison.com>";
      type="Application/X-FixedRecord"
      start-info="-o ps"
```

```
--example-1
```

```
Content-Type: Application/X-FixedRecord
Content-ID: <950120.aaCC@Xison.com>
```

```
25
10
34
10
25
21
26
10
```

```
--example-1
```

```
Content-Type: Application/octet-stream
Content-Description: The fixed length records
Content-Transfer-Encoding: base64
Content-ID: <950120.aaCB@Xison.com>
```

```
T2xkIE1hY0RvbmFsZCBoYWQgYSBmYXJtCkUgSS
BFIEkgTwpBbmQgb24gaGlzIGZhcm0gaGUgaGFk
IHNvbWUgZHVja3MKRSBJIEUgSSBPCldpdGggYS
BxdWFjayBxdWFjayBoZXJlLApIHF1YWNrIHF1
YWNrIHRoZXJlLApldmVyeSB3aGVyZSBhIHF1YW
NrIHF1YWNrCkUgSSBFIEkgTwo=
```

```
--example-1--
```

5.2 Text/X-Okie

The Text/X-Okie is an invented markup language permitting the inclusion of images with text. A feature of this example is the inclusion of two additional body parts, both picture. They are referred to internally by the encapsulated document via each picture's body part content-ID. Usage of "cid:", as in this example, may be useful for a variety of compound objects. It is not, however, a part of the Multipart/Related specification.

```
Content-Type: Multipart/Related; boundary=example-2;  
            start="<950118.AEBH@Xison.com>"  
            type="Text/x-Okie"
```

```
--example-2  
Content-Type: Text/x-Okie; charset=iso-8859-1;  
            declaration="<950118.AEB0@Xison.com>"  
Content-ID: <950118.AEBH@Xison.com>  
Content-Description: Document
```

```
{doc}  
This picture was taken by an automatic camera mounted ...  
{image file=cid:<950118.AECB@Xison.com>}  
{para}  
Now this is an enlargement of the area ...  
{image file=cid:<950118.AFDH@Xison.com>}  
{/doc}
```

```
--example-2  
Content-Type: image/jpeg  
Content-ID: <950118.AFDH@Xison.com>  
Content-Transfer-Encoding: BASE64  
Content-Description: Picture A
```

```
[encoded jpeg image]
```

```
--example-2  
Content-Type: image/jpeg  
Content-ID: <950118.AECB@Xison.com>  
Content-Transfer-Encoding: BASE64  
Content-Description: Picture B
```

```
[encoded jpeg image]
```

```
--example-2--
```

5.3 Content-Disposition

In the above example each image body part could also have a Content-Disposition header. For example,

```
...
--example-2
Content-Type: image/jpeg
Content-ID: <950118.AECB@XIson.com>
Content-Transfer-Encoding: BASE64
Content-Description: Picture B
Content-Disposition: INLINE

[encoded jpeg image]
--example-2--
```

User Agents that recognize Multipart/Related will ignore the Content-Disposition header's disposition type. Other User Agents will process the Multipart/Related as Multipart/Mixed and may make use of that header's information.

6. User Agent Requirements

User agents that do not recognize Multipart/Related shall, in accordance with [MIME], treat the entire entity as Multipart/Mixed. MIME User Agents that do recognize Multipart/Related entities but are unable to process the given type should give the user the option of suppressing the entire Multipart/Related body part shall be.

Existing MIME-capable mail user agents (MUAs) handle the existing media types in a straightforward manner. For discrete media types (e.g. text, image, etc.) the body of the entity can be directly passed to a display process. Similarly the existing composite subtypes can be reduced to handling one or more discrete types. Handling Multipart/Related differs in that processing cannot be reduced to handling the individual entities.

The following sections discuss what information the processing application requires.

It is possible that an application specific "receiving agent" will manipulate the entities for display prior to invoking actual application process. Okie, above, is an example of this; it may need a receiving agent to parse the document and substitute local file names for the originator's file names. Other applications may just require a table showing the correspondence between the local file names and the originator's. The receiving agent takes responsibility for such processing.

6.1 Data Requirements

MIME-capable mail user agents (MUAs) are required to provide the application:

- (a) the bodies of the MIME entities and the entity Content-* headers,
- (b) the parameters of the Multipart/Related Content-type header, and
- (c) the correspondence between each body's local file name, that body's header data, and, if present, the body part's content-ID.

6.2 Storing Multipart/Related Entities

The Multipart/Related media type will be used for objects that have internal linkages between the body parts. When the objects are stored the linkages may require processing by the application or its receiving agent.

6.3 Recursion

MIME is a recursive structure. Hence one must expect a Multipart/Related entity to contain other Multipart/Related entities. When a Multipart/Related entity is being processed for display or storage, any enclosed Multipart/Related entities shall be processed as though they were being stored.

6.4 Configuration Considerations

It is suggested that MUAs that use configuration mechanisms, see [CFG] for an example, refer to Multipart/Related as Multipart/Related/<type>, where <type> is the value of the "type" parameter.

7. Security considerations

Security considerations relevant to Multipart/Related are identical to those of the underlying content-type.

8. Acknowledgments

This proposal is the result of conversations the author has had with many people. In particular, Harald A. Alvestrand, James Clark, Charles Goldfarb, Gary Houston, Ned Freed, Ray Moody, and Don Stinchfield, provided both encouragement and invaluable help. The author, however, take full responsibility for all errors contained in this document.

9. References

- [822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", August 1982, University of Delaware, RFC 822.
- [CID] E. Levinson, J. Clark, "Message/External-Body Content-ID Access Type", 12/26/1995, RFC 1873
Levinson, E., "Message/External-Body Content-ID Access Type", February 1997, RFC 2111.
- [CFG] Borenstein, N., "A User Agent Configuration Mechanism For Multimedia Mail Format Information", September 23, 1993, RFC 1524
- [DISP] R. Troost, S. Dorner, "Communicating Presentation Information in Internet Messages: The Content-Disposition Header", June 7, 1995, RFC 1806
- [MIME] Borenstein, N. and Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies", June 1992, RFC 1341.

9. Author's Address

Edward Levinson
Xison, Inc.
47 Clive Street
Metuchen, NJ 08840-1060
USA
+1 908 549 3716
<Xison@cnj.digex.com>

