Network Working Group
Request for Comments: 1872
Category: Experimental

E. Levinson
Accurate Information Systems, Inc.

December 1995

The MIME Multipart/Related Content-type

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. This memo does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. 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.

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, a content-id. Start-info, a string or content-id list.
Encoding considerations:	Multipart content-types cannot have encodings.
Security considerations:	Depends solely on the referenced type.
Published specification:	This document.
Person & email address to contact for further information:	Edward Levinson Accurate Information Systems, Inc. 2 Industrial Way Eatontown, NJ 07724 +1 908 389 5550 +1 908 389 5556 (fax) ELevinson@Accurate.com

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-" header.

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.

Note: Constraining the "type" parameter's value to an existing media type allows the appropriate processing to be identified without creating yet another hierarchy of registered types. A possible default action would have the MIME mail User Agent (MUA) to display the "start" entity alone when it could process the media type as a basic type but not as Multipart/Related.

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 application processes first.

In the case of a Multipart/Alternative body part containing several entities with identical content-IDs the start entity should be selected using the Multipart/Alternative rules.

Note: The "start" parameter allows for types in which the root element gets generated by the sending application, perhaps on the fly. Such an application can create the "start" content-id when processing begins and then insert the body part when it is complete.

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

content-id 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. Examples

4.1 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 which the sender processes on the fly to generate the record length list. Consequently the list appears after the data.

```
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/octet-stream
Content-Description: The fixed length records
Content-Transfer-Encoding: base64
```

```
     Content-ID: <950120.aaCB@XIson.com>

     T2xkIE1hY0RvbmFsZCBoYWQgYSBmYXJtCkUgSS
     BFIEkgTwpBbmQgb24gaGlzIGZhcm0gaGUgaGFk
     IHNvbWUgZHVja3MKRSBJIEUgSSBPCldpdGggYS
     BxdWFjayBxdWFjayBoZXJlLAphIHF1YWNrIHF1
     YWNrIHRoZXJlLApldmVyeSB3aGVyZSBhIHF1YW
     NrIHF1YWNrCkUgSSBFIEkgTwo=
     --example-1
     Content-Type: Application/X-FixedRecord
     Content-ID: <950120.aaCC@XIson.com>

     25
     10
     34
     10
     25
     21
     26
     10
     --example-1--
```

4.2 Text/X-Okie

The Text/X-Okie is an invented markup language, similar to HTML, that permits 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-1--
```

5. 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 recognize Multipart/Related entities but are unable to process the given type shall either suppress the entire Multipart/Related body part or process the root alone. In either case the user should be notified of the MUA's action.

Handling Multipart/Related differs from other media types in that processing cannot be reduced to handling the individual entities. Existing media types are handled by MIME-capable MUAs handle in a straightforward manner. For basic media types (e.g., text, image, etc.) the body of the entity can be directly passed to a display process. Composite media types can be reduced to handing one or more discrete types.

Multipart/Related provides an irreducible composite media type.

The following sections discuss what information the processing application requires.

It is possible that an application specific "receiving agent" will manipulate the entities, after initial processing by the MIME User Agent, prior to invoking actual application process. From the viewpoint of the MUA, the receiving agent is the application. Okie, above, demonstrates 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 any for such processing.

5.1 Data Requirements

MIME-capable 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.

5.2 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.

5.3 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. It shall be the responsibility of the application handling the outermost Multipart/Related to insure the appropriate processing of embedded Multipart/Related entities.

5.5 Configuration Considerations

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

6. Security Considerations

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

7. Acknowledgments

This proposal is the result of conversations the author has had with many people. In particular, similar work was described by Harald A. Alvestrand (early drafts of Multipart/Related), Dave Crocker (Multipart/Families), and Keith Moore (Multipart/References). In addition, 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.

8. References

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

[CFG] Borenstein, N., "A User Agent Configuration Mechanism For Multimedia Mail Format Information", RFC 1524, Bellcore, September 1993.

[MIME] Borenstein, N. and 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.

9. Author's Address

Edward Levinson
Accurate Information Systems, Inc.
2 Industrial Way
Eatontown, NJ 07724-2265
USA

Phone: +1 908 389 5550
EMail: ELevinson@Accurate.com