Wrapping MIME Objects:  Application/MIME

STATUS OF THIS MEMO

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

To learn the current status of any Internet-Draft, please check the `1id-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

INTRODUCTION

MIME permits labeling and aggregating objects into complex hierarchies. Support for MIME mechanisms is not universal although it is growing quickly. Consequently gateways often are required to translate portions of a MIME object into its constituent pieces, as independent attachments. Further some uses of MIME's aggregation mechanism, Content-type Multipart, are leading to requirements for processing an entire aggregated object as a single unit. This is in contrast to Content-type Multipart/Mixed which specifies separate processing of constituent components, and to which multipart sub-types default if they are unknown to the processing software. This specification defines Content-type Application/MIME to provide an encapsulation mechanism for arbitrary MIME structures. This facilitates their treatment, as a single attachment, by software that is otherwise unfamiliar with the types of objects contained in the encapsulation. It is expected that Application/MIME will be especially useful in getting data past gateways.

The specification simply defines an object which contains MIME data. When the object is removed from its Content-type Application/MIME wrapper, what remains is an entire MIME
object, beginning with a (new) Content-type header.

APPLICATION/MIME SPECIFICATION

MIME type name: Application
MIME subtype name: MIME
Required parameters: <contained>
Optional parameters: None
Encoding considerations: May need BASE64 or QUOTED-PRINTABLE transfer encoding, as appropriate for the <contained> data. Careful use of QUOTED-PRINTABLE will maintain clear text as robust against gateway processing yet still be readable without special processing.

NOTE: items carried inside this MIME object must be fully conformant MIME structure and data; this includes all of the usual rules concerning network standard canonical representation.

Security considerations: See separate section in this document.

Published specification: See detail, below.

Rationale: Gateways and some other processing environments can alter and destroy MIME data structure; the defined data type will permit "hiding" the structure inside an object that is much less likely to be modified by such software. It also permits passing an aggregate object as a single entity, through processors that would otherwise separate the components.

Contact-info: See Contact section, below.

Detail specific to MIME-based usage:

This is a generic mechanism for encapsulating any MIME object structure. The object is self-defining, since it begins with a MIME Content-Type header and is then processed (recursively). The mechanism is intended for use when correct processing of the basic MIME structure is at risk, in effect allowing the MIME structure to be tunneled through such an environment.
The Content-Type parameter comprises the type and subtype tokens of the Content-Type header for the contained MIME component, i.e.,

\[
\text{contained} = \text{c-t} *(";" + \text{parameter})
\]

\[
\text{c-t} = "\text{Content-Type}=" + \text{type} + "/" + \text{subtype} + ""
\]

\[
\text{parameter} = \{\text{as defined for MIME}\}
\]

where the type and subtype tokens are defined by the MIME [2] specification. The value of the Content-Type parameter specifies the MIME Content-Type and subtype of the data structure contained inside the Application/MIME object.

The \text{<contained> parameters} replicate all of the parameters which are present in the Content-Type specification header referenced by the Content-Type parameter. That is, they replicate the contained object's parameters, in their entirety. This is done to facilitate dispatch of a particular type to a handler, without parsing the contained MIME structure.

IMPORTANT NOTE: If the contained object is, itself Application/MIME, then none of the \text{<contained> parameters} of that contained object shall be included.

(Using Application/MIME for doubly-wrapping MIME data may provide a necessary level of protection in some cases.)

GATEWAY AND PASS-THROUGH PROCESSING

Software which manipulates an Application/MIME component must do so ONLY when processing can be done fully and correctly. In the extreme, this may require full parsing of the contained MIME structure and its parameters, prior to deciding whether to take responsibility for the content. Typically, however, review of the Application/MIME type and \text{<contained> parameters} will suffice.

Any MIME-aware software which encounters an Application/MIME component must leave the component in its existing form, unless that software is able to fully and correctly process ALL of the component contents AND such processing is appropriate to the environment in which the software is operating.

IMPORTANT NOTE: Gateway implementors are specifically and strongly cautioned against modification of an application/mime component.

The question of whether to unwrap content which is embedded in an Application/MIME becomes very simple. Application/MIME is used to provide protection against mishandling by intermediaries. Hence, only end-system softwareâ€”including
gateways and regular email user agents should even consider touching the content and then it should only do so when the recipient has a basis for believing that the processing will be correct.

SIMPLE USAGE IN MIME-BASED EMAIL

This section is intended as a simple example of the gist of the formatting required to encapsulate MIME objects within Internet mail, using Application/MIME:

To:  
Subject:  
From:  
Date:  
Mime-Version: 1.0  
Content-Type: Application/MIME;  
content-type=Multipart/signed;  
protocol="application/somesigscheme";  
boundary="//signatureboundary//"

Content-Type: Multipart/Signed;  
boundary="//signatureboundary//";  
protocol="application/somesigscheme"

--//signatureboundary//
Content-type::<type of the user data>>

<<user data>>

--//signatureboundary//
Content-type:application/somesigscheme

<<signature control information>>

--//signatureboundary//--

DOUBLE-WRAPPED EXAMPLE

This section shows a contained object which is, itself, a contained object, double-wrapped for extra protection against decay:

To:  
Subject:  
From:  
Date:  
Mime-Version: 1.0  
Content-Type: Application/MIME;  
content-type=Multipart/signed;  
protocol="application/somesigscheme";  
boundary="//signatureboundary//"

Content-Type: Application/MIME;  
content-type=Multipart/signed;  
boundary="//signatureboundary//";  
protocol="application/somesigscheme"

Content-Type: Multipart/Signed;
SECURITY CONSIDERATIONS

MIME content often includes sensitive data, so that transmission often needs to attend to authentication, data integrity, privacy, access control, and the like as appropriate.

IMPORTANT NOTE: The recursive processing required by Application/MIME requires use of whatever security checks are applied to newly-received MIME data.

This specification does NOT, itself, provide any security-related mechanisms. As needed and appropriate, such mechanisms MUST be added, either via Internet MIME-based security services or any other services which are appropriate to the user requirements.

ACKNOWLEDGMENTS

The idea for Application/MIME first developed out of conversations with Einar Stefferud and Marshall Rose, in trying to find a way for exchanging valid Internet Mail, complete with RFC822 headers and MIME content, through environments that provided no other Internet Mail technology besides a gateway between the proprietary environment and the Internet. Additional benefits of this mechanism then surfaced during discussions on the S/MIME development list.

CONTACT

David H. Crocker <dcrocker@imc.org>
Internet Mail Consortium
675 Spruce Dr.
Sunnyvale, CA 94086 USA
Phone: +1 408 246 8253
Fax: +1 408 249 6205

Laurence Lundblade <lgl@qualcomm.com>
Qualcomm, Inc.
6455 Lusk Blvd
San Diego Ca 92121 USA
Phone: 619-658-3584

Jamie Zawinski <jwz@netscape.com>
Netscape Communications, Inc.
501 East Middlefield Road
Mountain View, CA 94043