MIME Content Type for BinHex encoded files

1. 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 at any timne. It is not appropriate to use
Internet Drafts as reference material or to cite them other than as
a "working draft" or "work in progress". Please check the
1idabstracts.txt listing contained in the internet-drafts Shadow
Directories on nic.ddn.mil, nnsc.nsf.net, nic.nordu.net,
ftp.nisc.sri.com, or munnari.oz.au to learn the current status of
any Internet Draft.

2. Abstract

This memo describes the format to use when sending BinHex4.0 files
via MIME [BORE93]. The format is compatible with existing
mechanisms for distributing Macintosh files. Only when available
software and/or user practice dictates, should this method be
employed. It is recommended to use application/applefile [FALT95]
for maximum interoperability.

3. Introduction

Files on the Macintosh consists of two parts, called forks:

DATA FORK: The actual data included in the file. The Data
                fork is typically the only meaningful part of a
                Macintosh file on a non-Macintosh computer system.
                For example, if a Macintosh user wants to send a
                file of data to a user on an IBM-PC, she would only
                send the Data fork.

RESOURCE FORK: Contains a collection of arbitrary attribute/value
                pairs, including program segments, icon bitmaps,
                and parametric values.

Additional information regarding Macintosh files is stored by the
Finder has in a hidden file, called the "Desktop Database".

Because of the complications in storing different parts of a
Macintosh file in a non-Macintosh filesystem that only handles
consecutive data in one part, it is common to convert the Macintosh
file into some other format before transferring it over the network.

AppleDouble file format [APPL90], encoded in MIME as
multipart/appledouble [FALT95] and application/applefile [FALT95] is
the preferred format for a Macintosh file that is to be included in
an Internet mail message, because it provides recipients with
Macintosh computers the entire document, including Icons and other
Macintosh specific information, while other users easily can extract
the Data fork (the actual data).

However, this specification provides for use of the currently
popular BinHex4.0 encoding schemes, as a convinience to the
installed base of users.

4. MIME format for BinHex4.0

MIME-base Apple information is specified by:

MIME type-name: APPLICATION
MIME subtype name: MAC-BINHEX40
Required parameters: none
Optional parameters: NAME, which must be a "value" as
                           defined in RFC-1521 [BORE93].
Encoding considerations: none
Security considerations: See separate section in the document
Published specification: Appendix A
Rationale: Permits MIME-based transmission of data
                           with Apple Macintosh file system specific
                           information using a currently popular,
                           though platform specific, format.

4a. Detail specific to MIME-based usage

Macintosh documents do not always need to be sent in a special
format. Those documents with well-known MIME types and
non-existent or trivial resource forks can be sent as regular
MIME body parts, without use of AppleSingle, AppleDouble or
BinHex4.0.

Documents which lack a data fork must be sent as AppleSingle
according to RFC 1740 [FALT95].

Unless there are strong reasons not to, all other documents
should be sent as AppleDouble according to RFC 1740 [FALT95].
This includes documents with non-trivial resource forks, and
documents without corresponding well-known MIME types.

It may be valuable in some cases to allow the user to choose one
format over another, either because he disagrees with the
implementor's definition of "trivial" resource forks, or for

reasons of his own.

        Only when available software and/or user practice dictates,
        should BinHex 4.0 be employed.

5. BinHex

   BinHex 4.0 is a propular means of encoding Macintosh files for
   archiving on non-Macintosh file systems and for transmission via
   Internet mail. (See Appendix A for a brief description of the
   BinHex 4.0 format.)

   The content-type application/mac-binhex40 indicates that the body of
   the mail is a BinHex4.0 file. Even though the BinHex encoding
   consists of characters which are not the same as those used in
   Base64 (those regarded as safe according to RFC-1521 [BORE93]) a
   transportation encoding should not be done.

   Even though a BinHex file includes the original Macintosh filename,
   it is recommended that a name parameter be included on the
   Content-Type header to give the recipient a hint as to what file is
   attached. The value of the name parameter must be a "value" as
   defined by RFC-1521 [BORE93]. Note that this restricts the value to
   seven-bit US-ASCII characters.

   5a. BinHex example

        Content-Type: application/mac-binhex40; name="car.hqx"

           [The BinHex4.0 file goes here]

6. References

   APPL90   AppleSingle/AppleDouble Formats for Foreign Files
            Developer's Note, Apple Computer, Inc., 1990

   FALT95   Faltstrom P., D. Crocker, and E. Fair, MacMIME: Description
            of the format to use when sending Macintosh files with MIME
            [BORE93], RFC 1740, Bunyip, UDEL, Apple, November 1995.

   BORE93   Borenstein N., and N. Freed, MIME (Multipurpose Internet
            Mail Extensions): Mechanisms for Specifying and Describing
            the Format of Internet Message Bodies, RFC 1521, Bellcore,
            Innosoft, September 1993.

7. Security Considerations

   To the extent that application/mac-binhex40 facilitates the

transmission of operating-system sensitive data, it may open a door
for easier relaxation of security rules than is intended either by

   the sender of the administrator of the sender's system.

8. **Acknowledgements**

   Thanks to all of the people on the ietf-822 list who have provided
   much meaningful input for this document. Some of them must though
   be remembered by name, because they have almost crushed my mailbox
   the last weeks with a very nice and interesting debate:

      Johan Berglund, Steve Dorner, David Gelhar, David Herron,
      Raymond Lau, Jamey Maze, John B. Melby, Jan Michael Rynning,
      Rens Troost, Peter Svanberg

9. **Authors Addresses**

Patrik Faltstrom
   Bunyip Information Systems inc.
   Suite 300
   310 Ste-Catherine St. West Montreal, Quebec
   CANADA H2X 2A1

   Email: paf@bunyip.com

   Dave Crocker
   Brandenburg Consulting
   675 Spruce. Dr.
   Sunnyvale, CA 94086
   USA

   Email: dcrocker@mordor.stanford.edu

   Erik E. Fair
   Engineering Computer Operations
   Apple Computer Inc.

   Email: fair@apple.com

Appendix A. **The BinHex format**

   Here is a description of the Hqx7 (7 bit format as implemented in
   BinHex 4.0) formats for Macintosh Application and File transfers.

   The main features of the format are:

   1) Error checking even using ASCII download
   2) Compression of repetitive characters
   3) 7 bit encoding for ASCII download

   The format is processed at three different levels:

      1) 8 bit encoding of the file:

   Byte: Length of FileName (1->63)
   Bytes: FileName ("Length" bytes)
   Byte: Version
   Long: Type
   Long: Creator
   Word: Flags (And $F800)
   Long: Length of Data Fork
   Long: Length of Resource Fork
   Word: CRC
   Bytes: Data Fork ("Data Length" bytes)
   Word: CRC
   Bytes: Resource Fork ("Rsrc Length" bytes)
   Word: CRC

      2) Compression of repetitive characters.

   ($90 is the marker, encoding is made for 3->255 characters)

   00 11 22 33 44 55 66 77   -> 00 11 22 33 44 55 66 77
   11 22 22 22 22 22 22 33   -> 11 22 90 06 33
   11 22 90 33 44   -> 11 22 90 00 33 44

   The whole file is considered as a stream of bits. This stream will
   be divided in blocks of 6 bits and then converted to one of 64
   characters contained in a table. The characters in this table have
   been chosen for maximum noise protection. The format will start
   with a ":" (first character on a line) and end with a ":".
   There will be a maximum of 64 characters on a line. It must be
   preceded, by this comment, starting in column 1 (it does not start
   in column 1 in this document):

    (This file must be converted with BinHex 4.0)

Any text before this comment is to be ignored.

The characters used is:

 !"#$%&'()*+,- 012345689@ABCDEFGHIJKLMNPQRSTUVXYZ[`abcdefhijklmpqr