

MIME (RFC2045-9)

- ◊ Multipurpose Internet Mail Extensions
 - Additions to allow
 - Non-ASCII characters
 - Multi-part messages
 - Structured messages
 - (almost anything)
- ◊ Before MIME
 - Just Send 8
 - Much-used solution to character set problem
 - But how is receiver to interpret the data?
- ◊ Always need to consider compatibility

Example Headers

```
Return-Path: <s4612099@maliwan.psu.ac.th>
Received: from maliwan.psu.ac.th (maliwan.psu.ac.th [202.12.73.4])
  by munnari.OZ.AU (8.12.8/8.11.6) with ESMTD id h6PEbr20026452
  for <kre@munnari.OZ.AU>; Sat, 26 Jul 2003 00:37:57 +1000 (EST)
Received: from Santichai (santichai.std.coe.psu.ac.th [172.30.20.134])
  by maliwan.psu.ac.th (8.11.6/8.11.6) with SMTP id h6PEbo104443
  for <kre@munnari.OZ.AU>; Fri, 25 Jul 2003 21:37:50 +0700 (ICT)
Message-ID: <000e01c352ba$54e9a9a0$86141eac@Santichai>
From: "Santichai Chuaywong" <s4612099@maliwan.psu.ac.th>
To: <kre@munnari.OZ.AU>
Subject: Some mistake about lecture slide
Date: Fri, 25 Jul 2003 21:37:52 +0700
MIME-Version: 1.0
Content-Type: multipart/alternative;
  boundary="====_NextPart_000_000B_01C352F5.009425C0"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.50.4807.1700
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4910.0300
```

- ◊ MIME-Version
 - Indicates MIME is in use
 - and which version
- ◊ Content-Type
 - What the message contains

MIME Specification

```
entity-headers := [ content CRLF ]
                 [ encoding CRLF ]
                 [ id CRLF ]
                 [ description CRLF ]
                 *( MIME-extension-field CRLF )
MIME-message-headers := entity-headers
                       fields
                       version CRLF
                       ; The ordering of the header
                       ; fields implied by this BNF
                       ; definition should be ignored.
MIME-part-headers := entity-headers
                    [ fields ]
                    ; Any field not beginning with
                    ; "content-" can have no defined
                    ; meaning and may be ignored.
                    ; The ordering of the header
                    ; fields implied by this BNF
                    ; definition should be ignored.
```

- ◊ Note, not quite ABNF
 - MIME predates ABNF Spec

MIME Specification

```
version := "MIME-Version" ":" 1*DIGIT "." 1*DIGIT
```

- ◊ From time when white space
 - And comments, line folding
- ◊ was implicit
 - could occur between any two tokens
- ◊ This caused problems
 - Implementations did not always work
 - Nor was it useful
- ◊ RFC2822 explicitly specifies white space
 - RFC2045 (etc) precede RFC2822...

```
MIME-Version: 1.(produced by MetaSend Vx.x)0
```

- Is OK
 - RFC2045 says so

```
MIME-Version: 1.2(produced by MetaSend Vx.x)0
```

- Is OK too
 - But unlikely to be processed correctly
- Not specified in RFC2045

MIME Content Type

```
content := "Content-Type" ":" type "/" subtype
          *("; parameter)
          ; Matching of media type and subtype
          ; is ALWAYS case-insensitive.
type := discrete-type / composite-type
discrete-type := "text" / "image" / "audio" / "video" /
                "application" / extension-token
composite-type := "message" / "multipart" / extension-token
extension-token := ietf-token / x-token
ietf-token := <An extension token defined by a
              standards-track RFC and registered
              with IANA.>
x-token := <The two characters "X-" or "x-" followed, with
           no intervening white space, by any token>
subtype := extension-token / iana-token
iana-token := <A publicly-defined extension token. Tokens
              of this form must be registered with IANA
              as specified in RFC 2048.>
parameter := attribute "=" value
attribute := token
            ; Matching of attributes
            ; is ALWAYS case-insensitive.
value := token / quoted-string
token := 1*<any (US-ASCII) CHAR except SPACE, CTLs,
         or tspecials>
```

Message Encoding

```
encoding := "Content-Transfer-Encoding" ":" mechanism
mechanism := "7bit" / "8bit" / "binary" /
            "quoted-printable" / "base64" /
            ietf-token / x-token
```

- ◊ Allows for different encodings of the same message
- ◊ Encoding is a defined format
 - Can be reversed when message arrives

```
Content-Type: text/plain; charset=ISO-8859-1
Content-transfer-encoding: base64
```

- ◊ Allows any data type
 - Including any (defined) character set
- ◊ In the message body

MIME Header Extensions

◇ RFC2047

```
encoded-word = "=?" charset "?" encoding "?"
                encoded-text "?="
charset = token      ; see section 3
encoding = token     ; see section 4
token = 1*<Any CHAR except SPACE, CTLs, and specials>
specials = "(" / ")" / "<" / ">" / "@" / "," / ";" / ":" /
           "/" / "[" / "]" / "?" / "." / "="
encoded-text = 1*<Any printable ASCII character other
               than "?" or SPACE>
               ; (but see "Use of encoded-words in message
               ; headers", section 5)
```

◇ Allows:

```
=?iso-8859-1?q?this=20is=20some=20text?=-
```

◇ Now any characters can occur in headers as well

- And be understood correctly

◇ Costs extra bytes in message

- Makes message hard to read without tools
- Nothing is free...

Multipart Messages

```
boundary := 0*69<bchars> bcharsnospace
bchars := bcharsnospace / " "
bcharsnospace := DIGIT / ALPHA / "'" / "(" / ")" /
                "+" / "-" / "_" / "." / "/" / ":" / "=" / "?"
dash-boundary := "--" boundary
                ; boundary taken from the value of
                ; boundary parameter of the
                ; Content-Type field.
multipart-body := [preamble CRLF]
                  dash-boundary transport-padding CRLF
                  body-part *encapsulation
                  close-delimiter transport-padding
                  [CRLF epilogue]
transport-padding := *LWSP-char
encapsulation := delimiter transport-padding
                 CRLF body-part
delimiter := CRLF dash-boundary
close-delimiter := delimiter "--"
preamble := discard-text
epilogue := discard-text
discard-text := *(*text CRLF) *text
body-part := MIME-part-headers [CRLF *OCTET]
OCTET := <any 0-255 octet value>
```

Multipart Example

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Date: Mon, 22 Mar 1993 09:41:09 -0800 (PST)
Subject: Formatted text mail
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary=boundary42

--boundary42
Content-Type: text/plain; charset=us-ascii
... plain text version of message goes here ...

--boundary42
Content-Type: text/enriched
... RFC1896 text/enriched version of same message ...

--boundary42
Content-Type: application/x-whatever
... fanciest version of same message goes here ...

--boundary42--
```