

# MIME 時代のメールの利用と管理について

山本和彦  
奈良先端科学技術大学院大学  
Kazu@Mew.org

中村素典  
京都大学  
motonori@econ.kyoto-u.ac.jp

## 概要

MIME の普及に伴い、さまざまな形式のデータの配送、多言語メッセージの交換、プライバシーの保護が可能となってきた。しかしながら、MIME の機能をいい加減に実装しているメールリーダーや不適切な処理を施す配送システムが存在するため、これらのサービスに悪影響を及ぼしている。本稿では、MIME 時代においてメールリーダーや配送システムが守るべき項目を、とくに日本語に注目して説明する。

## 1 はじめに

MIME[1] は、これまでのテキスト・メール [2] に代わるマルチメディア・メールであり、この数年で急速に普及した。MIME を利用すれば、テキスト以外のデータを添付できるだけでなく、さまざまな言語を含んだテキストの配送や、データのプライバシー保護が可能になる。これらの便利な機能を利用するには、メールリーダーが正しい書式を生成すること、および、中継システムが不適切な変換をしないことが必要である。しかしながら、正しい書式を生成できるメールリーダーはそれほど多くなく、また、不適切な変換を施す中継システムも少なからずあるのが現状である。そこで、本稿ではこれまでの経験から得られた知識をもとに、MIME 時代に守るべき項目を挙げ、正確な知識が普及するための一助としたい。

## 2 MIME

ヘッダに “MIME-Version: 1.0” というフィールドを持つメールが、MIME メールである。MIME の特徴は、以下の 4 つに集約される。

1. データ型を示すラベルの導入
2. 本文の構造化
3. 安全な符号化の定義

### 4. ヘッダの拡張

それぞれを以下の節で説明する。

#### 2.1 データ型を示すラベルの導入

Content-Type: (以下 CT: と略記) というフィールドによりデータ型を指示する。データ型は主データ型と副データ型をスラッシュで区切って指定する。主データ型には、Text、Application、Image、Audio、Video、Multipart、Message がある。それぞれのデータ型は、固有のパラメータを採ることが可能である。たとえば、Text には charset というパラメータが存在し、テキストに使われている文字コード (coded character set) を指定できる。このパラメータのおかげで、地域別のルールに困われることなく、さまざまな言語を配送できるようになった。

#### 2.2 本文の構造化

Multipart と Message は、本文を構造化するためのデータ型である。Multipart は、複数のデータを格納する。また、Message は、(テキスト、および、MIME の両方を含む) メールを MIME メールに格納する。このように、Multipart と Message を使えば、本文に再帰的な構造を与えられる。

## 2.3 安全な符号化の定義

長年、バイナリ・データなどを配送する場合には、uuencode によってデータを符号化していた。uuencode は 8 ビットで区別される文字 3 個 (24 ビット) を 6 ビットで区別される文字 4 個 (24 ビット) に変換する。しかし、6 ビット文字として安全ではない記号、たとえば空白やコンマが使われている。MIME では、より安全な記号を使う Base64 符号化方式を定めた。また、ヨーロッパ言語で書かれたテキストのように、大部分 ASCII であるが一部 ASCII ではない文字が現れるテキストを読める部分を多く残すように符号化する Quoted-Printable も定めている。両者は、Content-Transfer-Encoding: フィールド (以下 CTE: と略記) で指示する。

## 2.4 ヘッダの拡張

これまで、ヘッダには ASCII 以外の文字コードは格納できなかった。MIME では、Base64 と全く同じ B 符号化方式と Quoted-Printable の亜種である Q 符号化方式により、他の文字コードを ASCII 文字列に変換することで格納可能となる [3]。よって、現在では「Subject: に日本語を入れてはいけません」という必要はなくなった。(MIME のヘッダ拡張に対応していないメール・リーダを利用している場合は、ヘッダに日本語を入れてはならない。) Subject: のみならず、以下のように From: 行に日本語の名前を挿入してもよい。

```
From: K.Yamamoto
(=?ISO-2022-JP?B?GyRCQzNLXE9CSScbKEI=?=)
<Kazu@Mew.org>
```

MIME のヘッダ拡張に対応しているメールリーダで読めば、この例は以下のように表示される。

```
From: K.Yamamoto (山本和彦) <Kazu@Mew.org>
```

## 3 メールリーダの注意点

メールやネットニュースで日本語を配送するには、ISO-2022-JP[4] という文字コードを利用する必要がある。この節では、ISO-2022-JP を扱う上で発生する問題について説明する。

## 3.1 ISO-2022-JP

テキスト・データが ISO-2022-JP である場合、charset には ISO-2022-JP を指定する。また、ISO-2022-JP は 7 ビットであるので、CTE: には 7bit (無変換で 7 ビットに収まっているという意味) を指示する。以下に例を示す。

```
Content-Type: text/plain;
charset=iso-2022-jp
Content-Transfer-Encoding: 7bit
```

ここに日本語がくる

ISO-2022-JP では、ASCII、JIS X 0201 1976、JIS X 0208 1978、および、JIS X 0208 1983 という 4 種類の文字集合 (graphic character set) を格納できる。しかし、日本語のメールを書くときには ASCII と JIS X 0208 1983 のみを使用すべきである。理由は、署名の説明の際に詳しく述べる。

## 3.2 charset パラメータの間違い

charset の指定に関し間違いを犯すメールリーダが多い。

まず、英語用に作られたメールリーダをいい加減に日本語に拡張したメールリーダは、決め打ちで charset に ISO-2022-JP を指定する。つまり、本文に ASCII 文字しかなくても、ISO-2022-JP を指示してしまう。このようなメールリーダで英語のメールを書いて送ると、受信者のメールリーダに ISO-2022-JP の処理機能を要求していることになる。英語圏で利用されているメールリーダは、charset=US-ASCII しか処理できないものが多く、このメールを表示できない可能性は極めて高い。この問題を避けるためには、charset に最小限の文字コードを指定する必要がある。

つぎに、範囲であるが、charset の影響が及ぶのは直後のデータのみである。Multipart を使って複数のテキスト・データを格納する場合、それぞれのパートに含まれるテキスト・データに対し最小限の適切な charset を指定する必要がある。いい加減な実装では、後続するパートの charset に先行するパートで指定された charset の影響が及ぶ。

### 3.3 CTE: の間違い

ISO-2022-JP は 7 ビットであるため、全く符号化する必要はない。しかしながら、わざわざ Base64 や Quoted-Printable で符号化するメールリーダが存在する。MIME に対応していないメールリーダでは、このメールを復号化できない。よって、ISO-2022-JP を格納したテキスト・データを符号化するべきではない。

また、CTE: に符合化方式として 8bit を指定するメールリーダもある。確かに 8bit は無変換であるが、charset と同様に適切に 7bit を指示すべきである。

### 3.4 1 バイト・カナ

JIS X 0201 1978 では、8 ビット (最高位ビットが 1 の) 部分にカナを割り当てている。この 1 バイト・カナは、等幅フォントにおいて漢字を表現する全角文字の半分の幅に相当する半角文字で表示されることが多いため、俗に半角カナと呼ばれている。現在、1 バイト・カナを格納する文字コードの標準は存在しない。ISO-2022-JP のテキスト・データに 1 バイト・カナを格納するのは誤りである。それにもかかわらず、1 バイト・カナは以下のようなパターンで間違っ て ISO-2022-JP に挿入されることがある。

- 8 ビット部分に直接
- SO < 7 ビットに落した 1 バイト・カナ > SI
- SO < 8 ビットの 1 バイト・カナ > SI
- ESC ( I < 7 ビットに落した 1 バイト・カナ > ESC ( B
- ESC ) I < 8 ビットの 1 バイト・カナ >

1 バイト・カナを入力できないメールリーダでも誤って、1 バイト・カナを挿入してしまう可能性がある。端的な例は以下の 2 つである。

- 1 バイト・カナの混じったメールを引用する
- 1 バイト・カナの混じったメールを転送する

メールリーダには、このような場合に 1 バイト・カナを排除する機能が求められている。

### 3.5 補助漢字

JIS X 0212 1990、いわゆる、補助漢字には、JIS X 0208 1983 にはない漢字が定められており、これらの漢字の需要も少なからずある。現在、補助漢字を格納できる文字コードには、ISO-2022-JP-2 [5] がある。

ISO-2022-JP の改定版 [6] では、混乱を避けるために ISO-2022-JP というラベルの下での補助漢字の利用を認めていない。後述の JIS X 0213 への移行の際に混乱は生じると予測できるので、今混乱を生むような規格の変更は得策ではないと考えられる。

補助漢字の利用は意見の分かれるところであるが、現在では charset に ISO-2022-JP-2 を指定する方法が一番無難である。1 バイト・カナと同様に、ISO-2022-JP に誤って補助漢字が紛れ込む可能性がある。charset に ISO-2022-JP と指定するなら、補助漢字を排除しなければならない。

### 3.6 ヘッダ

いわゆるヘッダ部分、および、Multipart で複数のデータを格納する場合は各パートの MIME ヘッダ部分には、直接 ISO-2022-JP を挿入してはならない。ISO-2022-JP を格納する場合は、RFC 2047 に従って適切な符号化を施す必要がある。

MIME のヘッダ拡張の問題点は、RFC 1522 と RFC 2047 という 2 つの規格が存在することである。RFC 1522 はすでに破棄されているので、RFC 1522 しか実装していないメールリーダは、RFC 2047 を実装すべきである。

RFC 1468 には、ISO-2022-JP をヘッダに格納する場合は、B 符号化を用いよと書かれた。このため、B 符号化だけを実装し、Q 符号化を取り扱えないメールリーダが多い。しかしながら、メールリーダは、B 符号化のみならず Q 符号化も実装しなければならない [7]。

## 4 配送システムの注意点

この節では、配送システムの問題点について説明する。

## 4.1 多言語の配送

charset パラメータのおかげで、さまざまな言語によるメッセージを配送できるようになったが、配送システムが盲目的に文字コードを変換すると、メッセージが正しく伝わらなくなる。たとえば、第1パートに ISO-2022-JP、第2パートに ISO-8859-1 を格納したメールが、MIME を無視して ISO-2022-JP を Shift\_JIS に変換するシステムに送られると、ISO-8859-1 が誤ったデータに変換されてしまう。

一般的に、配送システムは文字コードを変換すべきではない。必要であれば、文字コードはメールリーダーで変換されるべきである。もし、配送システムで文字コードを変換しなければならないなら、charset パラメータを考慮する必要がある。

## 4.2 電子署名

現在では、PGP/MIME[8] や S/MIME[9] に基づいて電子署名を施すことにより、内容の信頼性を向上できる。電子署名では、1 ビットの変更も改竄とみなされるので、署名用の書式が厳密に定義されている [10]。日本語のテキストに署名する場合は、配送書式である ISO-2022-JP を使う必要がある。

前述のように、ISO-2022-JP は、ASCII、JIS X 0201 1976、JIS X 0208 1978、および、JIS X 0208 1983 が格納できる。EUC-JP と Shift\_JIS は、(定義は曖昧であるが) この4つの文字集合の内、2つのみしか格納できない(たとえば ASCII と JIS X 0208 1983)。よって、ASCII と JIS X 0201 1976、あるいは、JIS X 0208 1978 と JIS X 0208 1983 を同時に含む ISO-2022-JP が EUC-JP や Shift\_JIS に変換された場合、情報が欠落し元に戻せない。よって、署名の検証に失敗し、改竄とみなされる。

端的な例としては、最終の配送サーバがローカルのスプールに保存するために Shift\_JIS に変換することが挙げられる。また、おせっかいにも、ASCII と JIS X 0201 1976、あるいは、JIS X 0208 1978 と JIS X 0208 1983 のエスケープ・シーケンスを入れ換える配送システムが存在する。

前述のように、配送システムは文字コードを変換すべきではない。別の表現で言えば、配送システムは無変換でメールをスプールに蓄積するべきである。このため、メールリーダーは、直接 ISO-2022-JP を処理する機能を持たなければならない。また、ISO-

2022-JP では ASCII と JIS X 0208 1983 のみを使うようにすべきである。

## 4.3 8 ビット・クリーン

ヨーロッパ言語では、8 ビットの文字コードが利用されている。また、中国本土や韓国でも、8 ビットの文字コードが一般的になるようである。配送システムが 8 ビット・クリーンであれば、これらの文字コードも無変換、つまり CTE: 8bit で配送できても便利である。よって、MIME 時代の配送システムは 8 ビット・クリーンであるべきである。

MIME が普及するまでは、8 ビット・クリーンの配送システムが多かった。しかし、RFC 1652[11] で 8BITMIME が定義されたことに伴い、明示的に 8BITMIME と返答しない配送システムに対しては、7 ビットに符号化して送る配送システムが普及してしまった。また、8 ビットの文字コードによるメッセージが MIME に従っていない場合に、強制的に MIME 形式に変換するような設定を可能とする配送システムも存在している。これらは、MIME 導入時の失敗の 1 つと考えられる。

これからは、配送システムを 8 ビット・クリーンに戻し、余計な MIME 変換が行われないように設定する努力が必要である。普及度の高い Sendmail を利用しているシステムでは、以下に挙げる項目について確認しておくことが望ましい。

- sendmail 8.7 以降のバージョンを利用
- コンパイル時に MIME8TO7 を定義している (デフォルト)  
sendmail の起動時に -d0.4 スイッチを指定することで確認できる。定義されていない場合は、SMTP でメールを受信する際に 8BITMIME 応答を返さない。
- SevenBitInput オプションは False  
sendmail.cf に O SevenBitInput=True という行がないこと。True になっていると、8 ビット文字の最高位ビットが落とされる。デフォルトは False なので、宣言自体がなくてもよい。
- EightBitMode オプションは pass8  
sendmail.cf の O EightBitMode 行に pass8 以外の値が設定されていないこと。pass8 以外であ

ると、配信先が 8BITMIME 応答をするにもかかわらず MIME 変換が行われたり、8 ビット文字が含まれているメールの配信が拒否されることがある。デフォルトは pass8 なので、宣言自体がなくてもよい。CF で作成した sendmail.cf はデフォルトで値が mime になっている場合があるので注意が必要。

また、配信先の配送システムが、SMTP 接続時に 8BITMIME の応答を返さない場合であっても、メールを MIME に変換させないようにするためには、M 行で定義される smtp メーラのフラグ・リスト (F= の部分) にフラグ 8 を追加しておく。ただし、この場合、配信先側の配送システムで 8 ビット文字の最高位ビットが落とされる可能性があることに注意する必要がある。厳密には、フラグ 8 のが追加された smtp8 メーラとフラグ 8 のない smtp メーラの 2 種類を用意し、送り先ごとにメーラを選択するのが望ましい。ちなみに、このような設定は RFC 1652 に違反するものであり、配信先の配送システムを RFC 1652 に対応するように変更してもらうことが正しい対応である。

## 5 その他の技術的課題

これまでに述べた問題の他に、下記の技術的課題について説明する。

- MIME パラメータへの日本語
- JIS X 0213

### 5.1 MIME パラメータへの日本語

MIME のヘッダ拡張では、ヘッダ中に ASCII 以外の文字コードを格納する符号化が定義されていた。しかし、MIME パラメータの値として ASCII 以外の文字コードを指示する規格はなかった。これが不都合な例としては、添付したデータのファイル名として日本語が利用できないことが挙げられる。

最近、RFC 2184[12] で MIME パラメータの値として ASCII 以外の文字コードを指定する方式が定められた。そこで、日本語で利用する場合の書式を今後定める必要がある。メールを利用する場合、日本語はすべからず ISO-2022-JP を利用している。しか

し、ファイル名は EUC-JP や Shift\_JIS が利用されている場合が多いので、これらの文字コードの使用を検討する必要があるだろう。

### 5.2 JIS X 0213

JIS X 0213 は、現在策定が進められている次世代の日本語文字集合である。JIS X 0213 は、JIS X 0208 と JIS X 0212 を包含している。JIS X 0213 への移行が完了すれば、補助漢字の問題は解決できる。ただし、移行には多くの混乱が生じると予想される。

また、IETF ではインターネットの文字コードの標準に、Unicode(UTF-8)[13] を採択したので、こちらの動向にも目を光らせる必要がある。

## 6 おわりに

多言語の配送や電子署名の利用のためには、適切に実装されたメールリーダや配送システムの普及が不可欠である。もし、いい加減に実装されたメールリーダを使っている場合は、まずそのベンダーに修正するよう要求を出して頂きたい。それでもダメなら、現在では代替品はたくさんあるので、他のメールリーダに乗り換えることをお勧めする。いい加減に実装されているメールリーダを利用すること自体、他の人に迷惑を掛けていることを心に留めて頂きたい。

また、不必要に文字コードを変換したり、8 ビット・クリーンでない配送システムを利用している場合は、適切な配送システムの導入を検討して頂きたい。このような配送システムは、フリーで存在する。

## 参考文献

- [1] N. Freed and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [2] D. Crocker, "Standard for the format of ARPA Internet text messages", RFC 822, August 1982.
- [3] K. Moore, "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.
- [4] J. Murai, M. Crispin, and E. van der Poel, "Japanese Character Encoding for Internet Messages", RFC 1468, June 1993.

- [5] M. Ohta and K. Handa, “ISO-2022-JP-2: Multilingual Extension of ISO-2022-JP”, RFC 1554, December 1993.
- [6] E. Wada, J. Murai, and K. Yamamoto, “Japanese Character Encoding for Internet Messages”, Internet-Draft, November 1997.
- [7] N. Freed and N. Borenstein, “Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples”, RFC 2049, November 1996.
- [8] M. Elkins, “MIME Security with Pretty Good Privacy (PGP)”, RFC 2015, October 1996.
- [9] Steve Dusse, Paul Hoffman, Blake Ramsdell, Laurence Lundblade, and Lisa Repka, “S/MIME Message Specification”, Internet-Draft, October 1997.
- [10] J. Galvin, S. Murphy, S. Crocker, and N. Freed, “Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted”, RFC 1847, October 1995.
- [11] Klensin, N. Freed, M. Rose, E. Stefferud, and D. Crocker. “SMTP Service Extension for 8bit-MIMEtransport”, RFC 1652, July 1994.
- [12] N. Freed and K. Moore, “MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations”, RFC 2184 , August 1997.
- [13] F. Yergeau, “UTF-8, a transformation format of ISO 10646”, Internet-Draft, September 1997.