JP AAWG JAPAN ANTI-ABUSE WORKING GROUP

MIMEと文字コードの闇

Hornetsecurity株式会社
平野善隆

自己紹介

名前　　平野 善隆

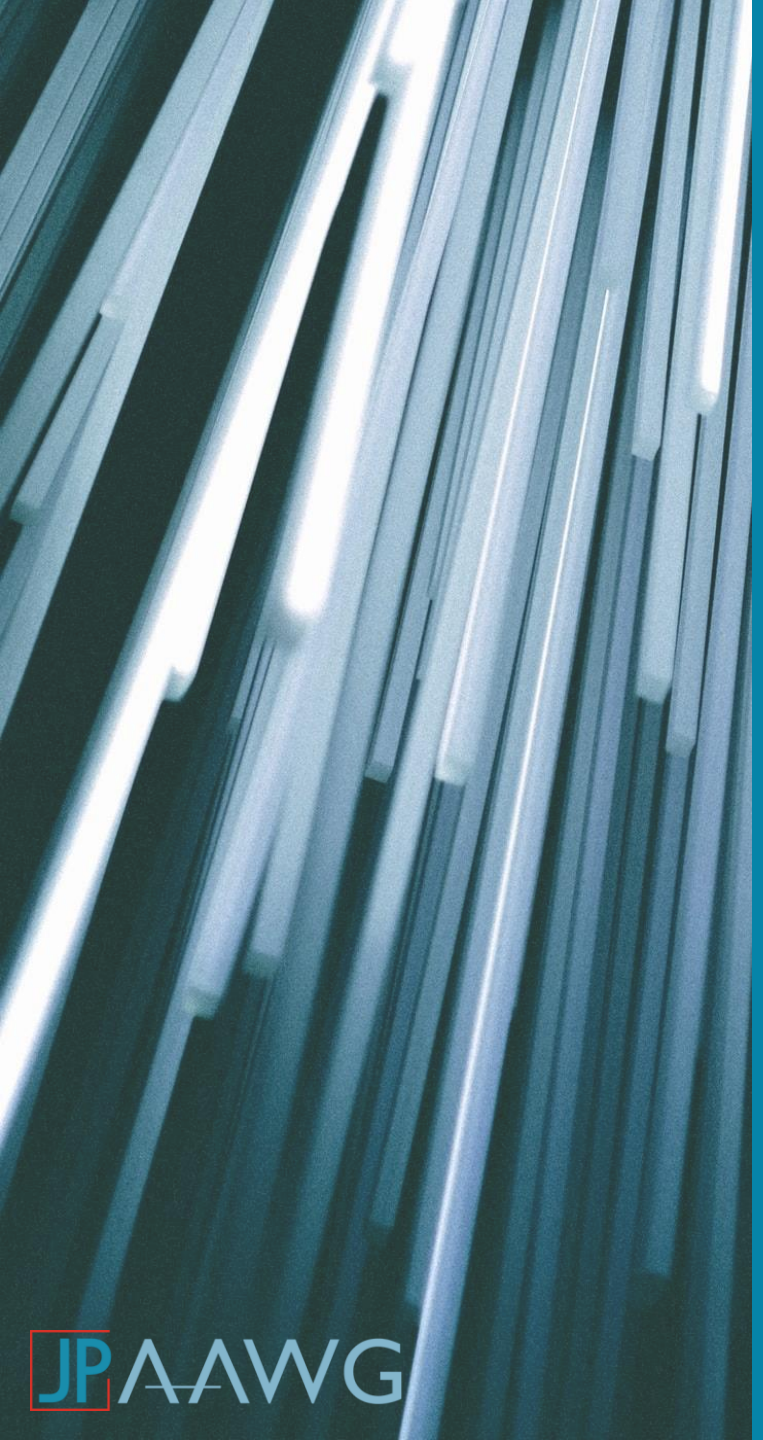
所属　　Hornetsecurity株式会社
(Vade Japan 株式会社)
Principal Messaging Engineer

文字コードとの関わり
1990年代前半に日本のPC通信上で
ハングルを扱う環境を開発。
この過程で様々な文字コードを変換。
日本語の文字コード変換も開発

趣味　　世界の長距離の自転車大会(1,200kmとか、 2,000kmとか)
バンド演奏
ドメイン調査

もくじ

- メールヘッダの読み方
- いにしえのインターネットでの日本語
- MIMEの登場
- 国際化ドメイン



基本的なヘッダ

JPAAWG

メールソースの例

```
MIME-Version: 1.0
Message-ID: <20250227132503000 1.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

ヘッダの基本

- ヘッダ名: 内容　で構成される
- 1つのヘッダは1行で表されている
- 空行の前までがヘッダ

```
MIME-Version: 1.0
Message-ID: <20250227132503000 1.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

ここは本文

Message-ID

- このメールを特定する世界で1つのID
- <○○@○○>の形式
- @の後ろはドメインが推奨される

```
MIME-Version: 1.0
Message-ID: <20250227132503000l.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

Date

- メールが送信された日時
- 曜日, 日 月 年 時:分:秒 タイムゾーン の形式
- 曜日はオプション

```
MIME-Version: 1.0
Message-ID: <20250227132503000l.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

From

- メールを送信した人
- 形式は display name <メールアドレス>　など
- envelope fromと同じである必要はない

```
MIME-Version: 1.0
Message-ID: <20250227132503000l.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

To

- メールの送信先
- 形式は display name <メールアドレス>　など
- envelope toと同じである必要はない

```
MIME-Version: 1.0
Message-ID: <20250227132503000 1.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

From, To, Cc メールアドレスの形式

- To: hirano@hornetsecurity.com
- To: <hirano@hornetsecurity.com>
- To: HIRANO <hirano@hornetsecurity.com>

など

Display Name
と呼びます

Subject

- 件名

```
MIME-Version: 1.0
Message-ID: <20250227132503000l.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

本文

- CR/LF/CR/LFの後がBody

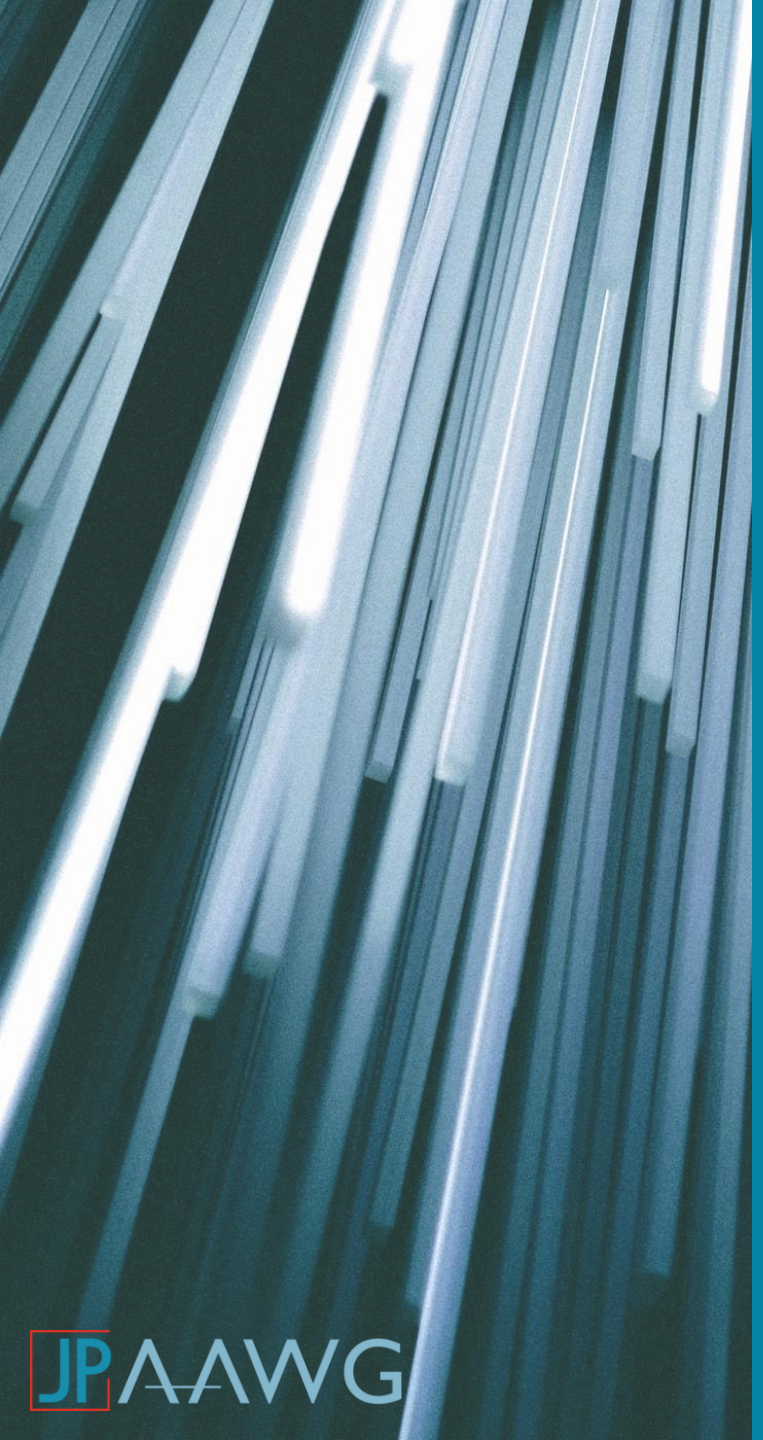
```
MIME-Version: 1.0
Message-ID: <20250227132503000 1.00002D14.0756@hornetsecurity.com>
Date: Thu, 27 Feb 2025 13:25:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

ヘッダの折り返し

- SMTPの1行の制限 ＝ 998バイト
- 78文字以下が望ましい
- 長いヘッダは折り返す
- 2行目以降はスペースやタブで始まる
- 改行、スペースやタブの連続は
  合わせて1つのスペースとみなされる

スペース →
TAB →

```
Subject: long long long long long long long long long long long long
 long long long subject
To: a@example.jp, b@example.jp,
    c@example.jp, d@example.jp
```



インターネットは7bit

JPAAWG

日本語の扱い

- 7bitしか通らないので、7bitの文字コードを使用する

```
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: HIRANO Yoshitaka <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: ここは日本語です

ここは日本語です
```

ここは日本語です = ESCB3$3$OF|K\8lG9ESC(B

アルファベットの文字コード(USASCII)

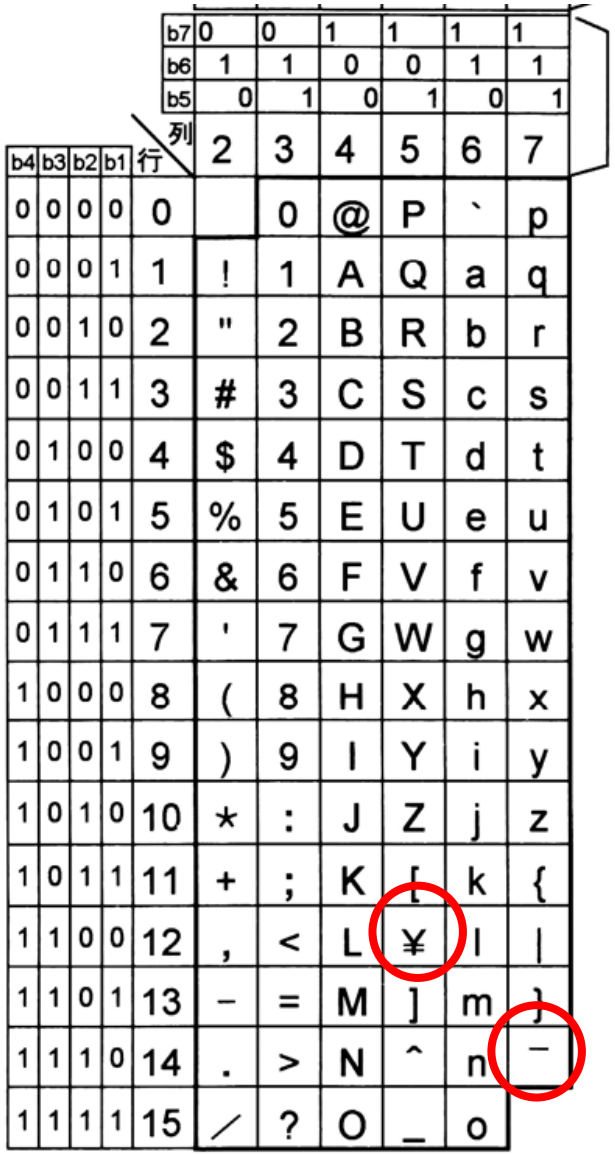
$b_7 b_6 b_5$ →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits / Column → / Row ↓	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	—	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

JIS X 0208

	00-1F	20	21-7E	7F	80-FF
00-1F	未使用	未使用	未使用	未使用	未使用
20	未使用	未使用	未使用	未使用	未使用
21-7E	未使用	未使用	ここだけ	未使用	未使用
7F	未使用	未使用	未使用	未使用	未使用
80-FF	未使用	未使用	未使用	未使用	未使用

94 x 94 = 8836文字

JIS X 0201



b7	0	0	1	1	1	1
b6	1	1	0	0	1	1
b5	0	1	0	1	0	1
b4 b3 b2 b1 行 \ 列	2	3	4	5	6	7
0 0 0 0 0		0	@	P	`	p
0 0 0 1 1	!	1	A	Q	a	q
0 0 1 0 2	"	2	B	R	b	r
0 0 1 1 3	#	3	C	S	c	s
0 1 0 0 4	$	4	D	T	d	t
0 1 0 1 5	%	5	E	U	e	u
0 1 1 0 6	&	6	F	V	f	v
0 1 1 1 7	'	7	G	W	g	w
1 0 0 0 8	(	8	H	X	h	x
1 0 0 1 9	)	9	I	Y	i	y
1 0 1 0 10	*	:	J	Z	j	z
1 0 1 1 11	+	;	K	[	k	{
1 1 0 0 12	,	<	L	¥	l	|
1 1 0 1 13	-	=	M	]	m	}
1 1 1 0 14	.	>	N	^	n	‾
1 1 1 1 15	/	?	O	_	o	

ISO-2022-JP

- ESC (B ➔ ASCII
- ESC (J ➔ JIS X 0201
- ESC $ @ ➔ JIS X 0208-1978
- ESC $ B ➔ JIS X 0208-1983 or 1990

ここは日本語です = ESCB3$3$OF|K\8lG9ESC(B

日本語の文字コード

- JIS X 0208 ≈ ISO-2022-JP
  - 7bitの文字コード
- EUC-JP
  - JIS X 0208の8bit目を立てたもの
- SHIFT_JIS
  - JIS X 0208をいい感じに移動したもの
  - 1byte目: 81～9F, E0～EF
  - 2byte目: 40～7E, 80～FC

EUC-JP

1バイト目	2バイト目	文字集合	内容
00-7F	-	ASCII (7bit)	ASCII (7bit)
A1-FE	A1-FE	JIS X 0208	JIS X 0208の8bitを立てたもの
8E	A1-DF	JIS X 0201 (半角かな)	半角かな
8F	A1-FE	JIS X 0212 (補助漢字)	拡張漢字

Shift_JIS

1バイト目	2バイト目	文字集合	内容
00-7F	-	ASCII (7bit)	ASCII (7bit)
A1-DF	-	JIS X 0201 (半角かな)	半角かな
80-9F, E0-EF	40-7E, 80-FC	JIS X 0208	JISX0208を移動したもの
F0-FC	40-7E, 80-FC	闇の部分	WindowsのShift_JISはここを含む

え、ほかにも?

- ISO-2022-JP-2
- ISO-2022-JP-3
- MS932
- CP932

- ・・・見なかったことにしましょう

しかし現実は

- 件名にShift_JISがそのまま記述される
  - 特にスパムメール

一方韓国では?

- EUC-KR (ks_c_5601)が主流

MIMEの登場

JPAAWG

MIME-Version

- ASCII以外のヘッダや本文を扱えるようにする
- 添付ファイルを扱えるようにする
- Content-○○ヘッダを使えるようにする

```
MIME-Version: 1.0
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

Content-Type

- Bodyがどんな形式かを表す
- MIME Type/MIME SubType; attribute=value のような形式

```
MIME-Version: 1.0
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: HIRANO Yoshitaka <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

Content-Transfer-Encoding

- Bodyのエンコード方式を表す
- 7bit, 8bit, base64, quoted-printable など

```
MIME-Version: 1.0
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: HIRANO Yoshitaka <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: test
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit

test
```

日本語の扱い

- MIMEエンコードを利用する

```
MIME-Version: 1.0
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: HIRANO Yoshitaka <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: =?ISO-2022-JP?B?GyRCJDMkMyRPRnxLXDhsJEckORsoQg==?=
Content-Type: text/plain; charset=iso-2022-jp
Content-Transfer-Encoding: 7bit

ここは日本語です
```

ここは日本語です = ESCB3$3$OF|K\8lG9ESC(B

ヘッダのエンコード

JPAAWG

ヘッダの日本語の扱い

- MIMEエンコードを利用する
- =?文字コード?BまたはQ?エンコードされた文字列?=

=?ISO-2022-JP?B?GyRCJDMkMyRPRnxLXDhsJEckORsoQg==?=

文字コード

= ESCB3$3$OF|K\8lG9ESC(B

=ここは日本語です

B: BASE64
Q: Quoted Printable

BASE64とは

- あいう
- E3 81 82 E3 81 84 E3 81 86

- 11100011 10000001 10000010 11000011 10000001
  10000100 11100011 10000001 10000110

- 11100011 10000001 10000010 11000011 10000001
  10000100 11100011 10000001 10000110

- 38 38 06 02 30 38 06 04 38 38 06 06

BASE64とは

- 38 38 06 02 30 38 06 04 38 38 06 06
- 56 56 6 2 48 56 6 4 56 56 6 6
- 44GCw4GE44GG

桁が4の倍数に
ならない場合は=を追加する

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

ヘッダの日本語の扱い (BASE64)

- MIMEエンコードを利用する
- BASE64の場合

=?ISO-2022-JP?B?GyRCJDMkMyRPRnxLXDhsJEckORsoQg==?=

文字コード　⬆ B: BASE64

= ESCB3$3$OF|K\8lG9ESC(B
=ここは日本語です

GyRCJDMkMyRP ...
6 50 17 2 9 3 12 36 12 50 17 15 ... ← BASE64の表を見ながら数字に直す(10進数)
06 32 11 02 09 03 0c 24 0c 32 11 0f ... ← 上の16進数表記　6bitの2進数表記↴
000110 110010 010001 000010 001001 000011 001100 100100 001100 110010 010001 001111 ...
00011011 00100100 01000010 00100100 00110011 00100100 00110011 00100100 01001111 ...
1b 24 42 24 33 24 33 4f ... ← 16進数表記　⇧8bitで区切り直す
ESC $ B $ 3 $ 3 O ...

Quoted Printableとは

- あいう
- E3 81 82 E3 81 84 E3 81 86

- =E3=81=82=E3=81=84=E3=81=86

- あabc
- =E3=81=82abc

ヘッダの日本語の扱い (Quoted Printable)

- MIMEエンコードを利用する
- Quoted Printableの場合

=?ISO-2022-JP?Q?=1BB3$3$OF|K\8lG9=1B(B?=

文字コード

= ESCB3$3$OF|K\8lG9ESC(B

=ここは日本語です

B: BASE64
Q: Quoted Printable

“”で囲むかどうか

```
1 =?ISO-2022-JP?B?GyRCSj9MbiRHJDkbKEI=?= <hirano@example.jp>
2 "=?ISO-2022-JP?B?GyRCSj9MbiRHJDkbKEI=?=" <hirano@example.jp>
```

RFC2047

An 'encoded-word' MUST NOT appear within a 'quoted-string'.

本当は囲んではいけない

でも、わりと囲まれている

スペースの扱い

- =?ISO-2022-JP?B?GyRCJCIbKEI=?==?ISO-2022-JP?B?GyRCJCQbKEI=?= ➔ あい
- =?ISO-2022-JP?B?GyRCJCIbKEI=?=a=?ISO-2022-JP?B?GyRCJCQbKEI=?= ➔ あaい
- =?ISO-2022-JP?B?GyRCJCIbKEI=?= a =?ISO-2022-JP?B?GyRCJCQbKEI=?= ➔ あ a い

- =?ISO-2022-JP?B?GyRCJCIbKEI=?= =?ISO-2022-JP?B?GyRCJCQbKEI=?= ➔ あい
- =?ISO-2022-JP?B?GyRCJCIbKEI=?= =?ISO-2022-JP?B?GyRCJCQbKEI=?= ➔ あい
- =?ISO-2022-JP?B?GyRCJCIbKEI=?=
  =?ISO-2022-JP?B?GyRCJCQbKEI=?= ➔ あい

- =?ISO-2022-JP?B?GyRCJCIbKEIgICAgIBskQiQkGyhC?= ➔ あ い

セキュリティリスク

- もし、先に文字コードをdecodeしてしまうと

=?utf-8?B?PGJhZGd1eUBiYWQuZXhhbXBsZS5jb20+LA== <me@vadesecure.com>

➔ <badguy@bad.example.jp>, <me@vadesecure.com>

➔ [
{displayName="", addr="badguy@bad.example.jp"},
{displayName="", addr="me@vadesecure.com"}
]

本文の
エンコーディング

JPAAWG

本文の日本語の扱い

- MIMEエンコードを利用する
- Content-Typeで文字コードを指定

```
MIME-Version: 1.0
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: HIRANO Yoshitaka <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: Test
Content-Type: text/plain; charset="iso-2022-jp"
Content-Transfer-Encoding: 7bit

ここは日本語です
```

ここは日本語です = ESCB3$3$OF|K\8lG9ESC(B

本文の日本語の扱い (BASE64)

- 本文にもMIMEエンコードを利用する

```
MIME-Version: 1.0
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: HIRANO Yoshitaka <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: Test
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: base64

44GT44GT44Gv5pel5pys6Kqe44Gn44GZ44CCDQo=
```

HTMLの文字コード

- HTMLも文字コードを指定できる

```
MIME-Version: 1.0
Message-ID: <20191114150001.00002D14.0756@hornetsecurity.com>
Date: Thu, 14 Nov 2019 15:00:01 +0900
From: HIRANO Yoshitaka <hirano@hornetsecurity.com>
To: <hirano@hornetsecurity.com>
Subject: Test
Content-Type: text/html; charset="iso-2022-jp"
Content-Transfer-Encoding: quoted-printable

<html lang=3D"ja">
<head>
  <meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Diso-2022-jp">
  <title>=1B$B$3$3$OF|K\8l$G$9=1B(B</title>
```

=1B ← ESC　=3D ← =

現実のHTMLメールの闇 ①

```
Content-Type: text/html; charset="ISO-2022-JP"
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang=3D"ja">

<head>
  <meta http-equiv=3D"Content-Language" content=3D"ja">
  <meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Dshi=
ft_jis">
  <title>=1B$B$8$c$i$sM7$S!&BN83=1B(B =1B$BNW;~A}4)9f=1B(B</title>
```

現実のHTMLメールの闇 ②

```
Content-Type: text/html; charset="euc-jp"
Content-Transfer-Encoding: 8bit

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-2022-jp"
/>
<title></title>
</head>
<body>
<font size="-1">平野 善隆 様 ( hirano@example.jp )</font>
```

EUC-JP

URLエンコード

- あいう
- E3 81 82 E3 81 84 E3 81 86

- %E3%81%82%E3%81%84%E3%81%86

- あabc
- %E3%81%82abc

URLエンコード + HTMLメール

```
Content-Type: text/html; charset="iso-2022-jp"
Content-Transfer-Encoding: quoted-printable

<html lang=3D"ja">
<head>
  <meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Diso-2022-jp">
  <title>=1B$B$3$3$OF|K\8l$G$9=1B(B</title>
</head>
<body>
  <a href=3D"https://example.jp/?name=3D%E5%B9%B3%E9%87%8E">Click here</a>
</body>
```

URLエンコード
UTF-8

メール作成時の文字コード選択・判定

- メール作成時に文字コードは自動判定される
- こんにちは ➔ iso-2022-jp
- hello! ➔ us-ascii
- 中国 ➔ iso-2022-jp? gb2312?

実際の文字化けの例

伊藤

お世になっております。 [redacted] のS原でございます。

素人の私には何が起きているのか皆目当がつきませんが、
最初にいただいたテキストを改めてこちらのメ`ルにdいたします。
}解Qの一助になれば幸いです。

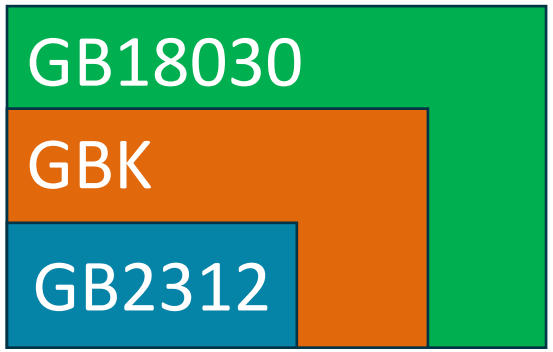
ご_Jの程よろしくおいいたします。

Vadeの高度なAIエンジンは、~えgなく浠しAける新しいデジタルな{威とメ`ルセキュリティの{威からITとビジネスおよび客を保oします。 スピアフィッシング、マルウェア、ランサムウェアなどからの保oを化します。

完全自踊するより、最後は手嬰扦g施が望ましいと思います。そして、その後にY果を_Jし、手嬰亲鳏盲郡猡韦私づけられるかどうかを考えて自踊すべきで、ほんとにojな努力をしてると思います。

なぜ文字化けしたのか

- 日本語のメール(UTF-8)に返信
- 中国語だと判定したのか、Content-TypeをGB2312に設定して送信
- しかし実際に送られた本文はGBK
- 受信した環境はGB2312の範囲外の文字を削除
- 文字化け



添付ファイル

JP AAWG

添付ファイル名のエンコード①

```
Content-Type: application/pdf;
        name="=?ISO-2022-JP?B?GyRCJS8lbCU4JUMlSCUrITwlSUxAOlkbKEIwMS5wZGY=?=";
        charset=ISO-2022-JP
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
        filename="=?ISO-2022-JP?B?GyRCJS8lbCU4JUMlSCUrITwlSUxAOlkbKEIwMS5wZGY=?="
```

クレジットカード明細01.pdf

添付ファイル名のエンコード②

RFC2231

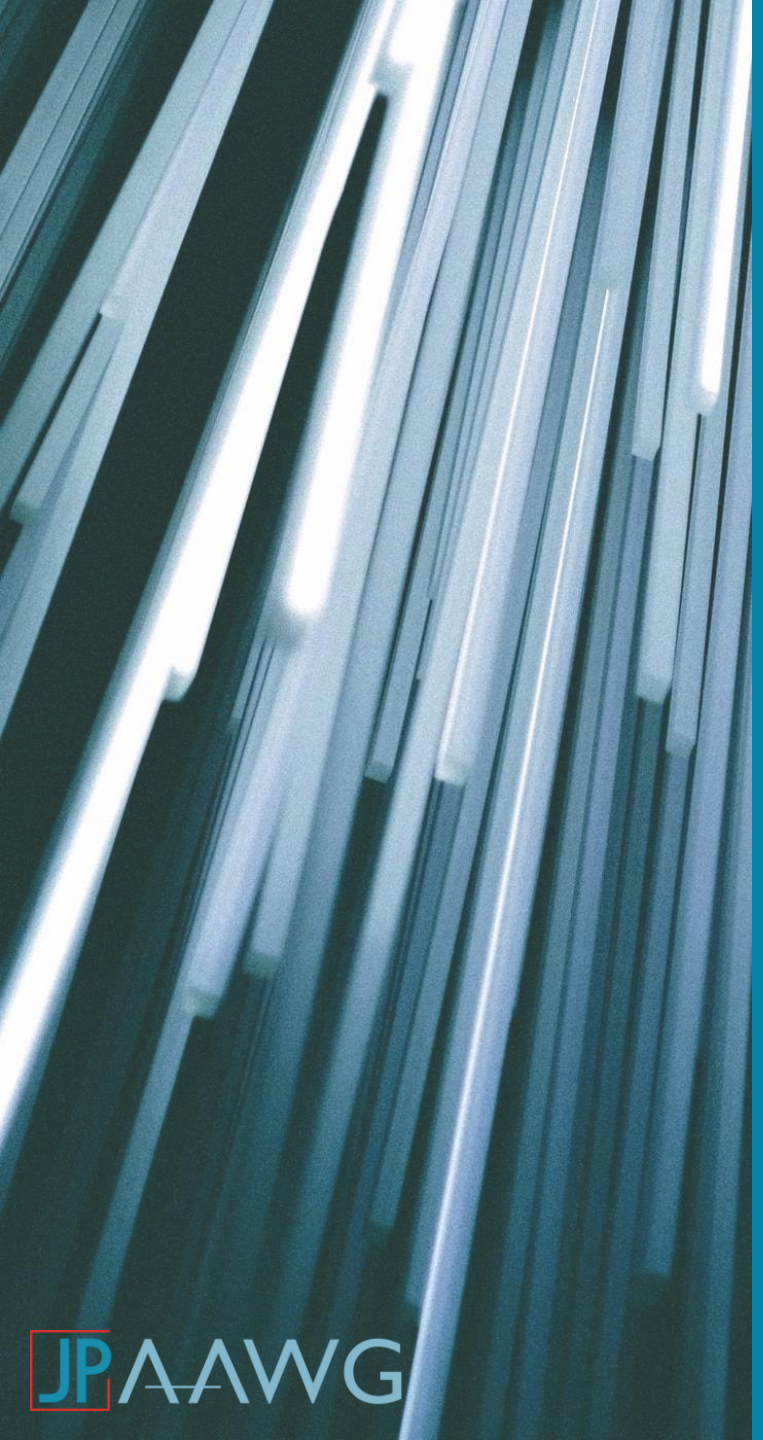
```
Content-Type: application/vnd.ms-excel;
 name="=?ISO-2022-JP?B?GyRCOk5FQEQiSTw8QTU/MX5FehsoQl8wNDAyMTJfMDEoGyRCSj9MbhsoQikueGxz?="
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
 filename*0*=ISO-2022-JP''%1B$B%3ANE%40D%22I%3C%3CA5%3F1~Ez%1B%28B;
 filename*1*=_040212_01%28;
 filename*2*=%1B$BJ%3FLn%1B%28B;
 filename*3*=%29.xls
```

採点帳票質疑応答_040212_01(平野).xls

添付ファイルのエンコード

- BASE64でエンコード

```
Content-Type: application/pdf;
        name="=?ISO-2022-JP?B?GyRCJS8lbCU4JUMlSCUrITwlSUxAOlkbKEIwMS5wZGY=?=";
        charset=ISO-2022-JP
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
        filename="=?ISO-2022-JP?B?GyRCJS8lbCU4JUMlSCUrITwlSUxAOlkbKEIwMS5wZGY=?="
```



UTF-7

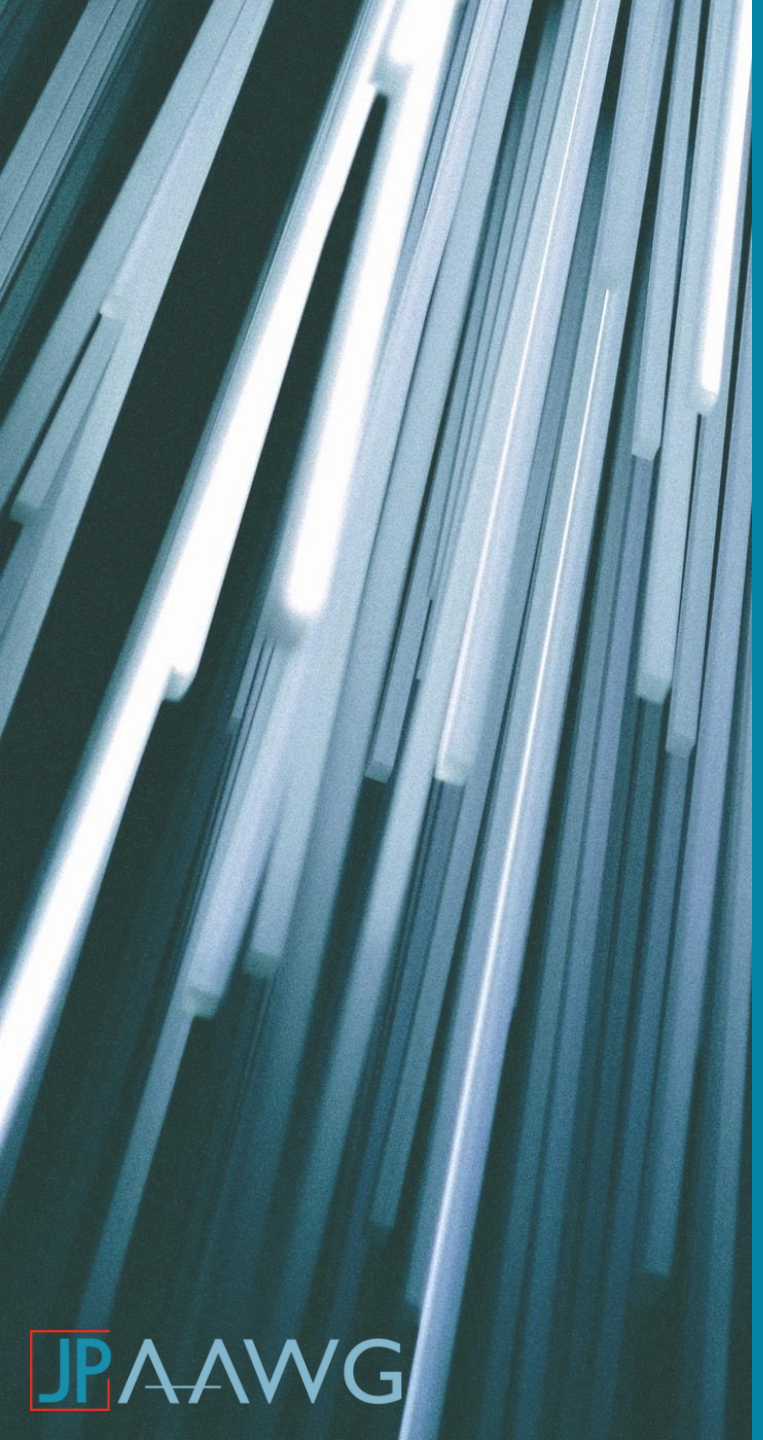
JPAAWG

IMAP フォルダ名のエンコード

修正UTF-7, mUTF-7

- 「&」以外の印字可能なUS-ASCII文字は必ずそのまま表記する。
- それ以外の文字はUTF-16のビッグエンディアンで符号化し、修正BASE64で符号化する。
- BASE64の文字の前に「&」後ろに「-」を置く。
- 「&」の文字自体は「&-」で表現する。

folder ➔ folder
こんにちはabc ➔ &MFMwkzBrMGEwbw-abc



UTF-8は万能なのか?

JP AAWG

UTF-8の仕組み

バイト数	1バイト目	2バイト目	3バイト目	4バイト目
1	00 – 7F			
2	C2 – DF	80 – BF		
3	E0 – EF	80 – BF	80 – BF	
4	F0 – F4	80 – BF	80 – BF	80 – BF

- 可変長
- ASCIIの範囲はそのまま表現される

日本語メールを処理するとは?

- 日本語のメールが表示できる

- 日本語のメールを検索できる
- 日本語のメールをフィルタできる

検索・フィルタでの問題点

- iso-2022-jp
- ESCB3$3$OF|K¥8lG9ESC(B (こんにちは)

$9で検索

- Shift_JIS
- 83 74 83 42 83 8b 83 5e (フィルタ)

Bで検索

- EUC-JP
- A4 B3 A4 B3 (ここ)

海(B3 A4)で検索

UTF-8にはこの問題がない

正規化

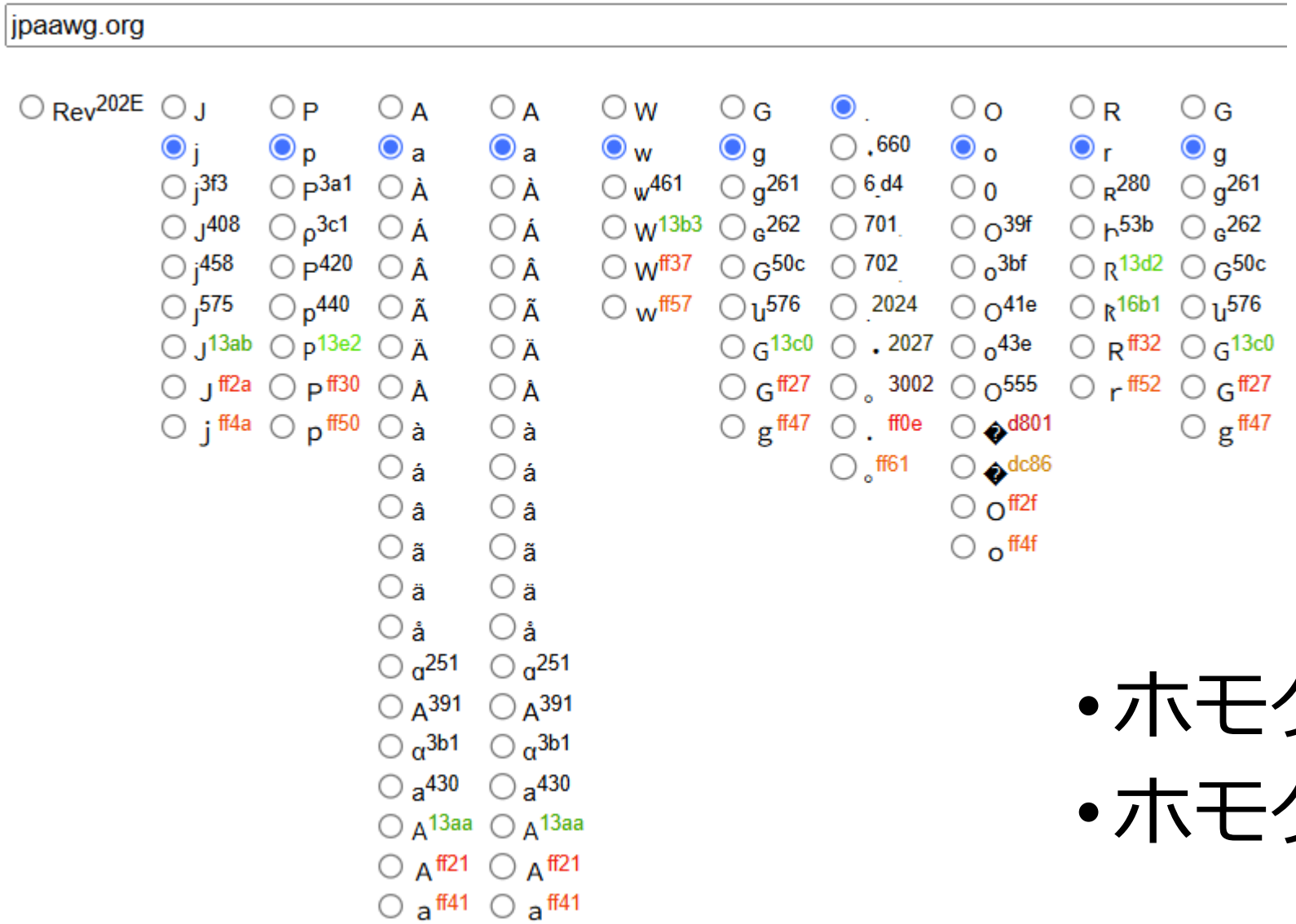
- 「が」(Windows) ≠ 「が」(MAC)
- 「e3 81 8c」≠ 「e3 81 8b e3 82 99」
- 「が」≠ 「か゛」

NFC: 結合可能な文字を可能な限り1つの文字に合成する
（例: é は e + ´ ではなく é で表現）
NFD: 可能な限り文字を分解する（例: é は e + ´ に分解）

NFKC: NFCに加えて互換文字（例: ½ → 1/2）も正規化
NFKD: NFD + 互換文字の分解

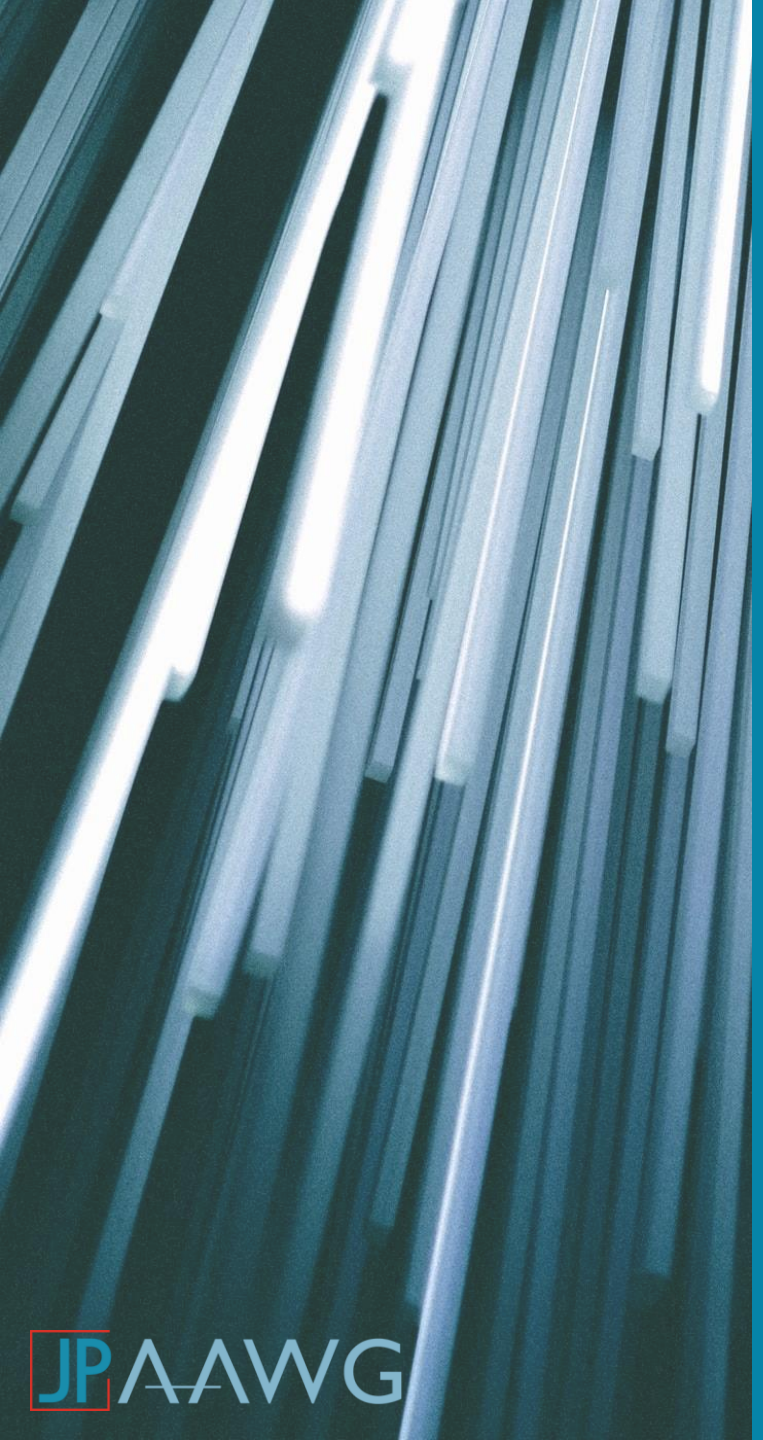
https://unicode.org/reports/tr15/

Homoglyph Attack Generator



- ホモグリフ
- ホモグラフドメイン

https://www.irongeek.com/homoglyph-attack-generator.php



メールアドレスの多言語化

JPAAWG

国際化ドメイン(IDN)のエンコード

- Punycodeでエンコードする
- ドメインのみ利用可能
- 既存システムがそのまま使える

hirano@東京.日本

➔ `hirano@xn--djrs72d.xn--wgv71a`

NFCで正規化する

EAIとは

- Email Address Internationalization
- メールアドレスのLocal Partも多言語化
- UTF-8を直接使う
- ドメイン部分もUTF-8で表記し、裏でpunycodeを使用
- 既存のプロトコルに拡張が必要
- RFC5335

平野@東京.日本

NFCでの正規化が推奨 (必須とは言っていない)
the use of normalization form NFC is RECOMMENDED.

EAIとホモグラフドメイン

- hirano@jpaawg.org

  ➔ hirano@xn--j-s1a96hyma80a.org

- hirano@jpaawg.org

  ➔ hirano@jpaawg.org

まとめ

JP AAWG

まとめ

- MIMEが普及する前は混沌としていた
- MIME普及後ましにはなったが、依然として様々な文字コードが混在している
- 例えUTF-8に統一したとしても問題はある
- 国際化ドメインやEAIなどは、危険な香りしかしない