

ICS 35.240.20
L 67



中华人民共和国国家标准

GB/T 18304—2001

信息技术 因特网中文规范 电子邮件传送格式

**Information technology—Chinese specification of internet
character transfer format for email**

2001-01-02 发布

2001-08-01 实施

国家质量技术监督局 发布

前 言

本标准规定了以 **GB 13000** 为基础在因特网电子邮件中传输中文字符和其他文种字符的消息格式。

本标准文本主要包括定义、通用电子邮件传输格式、**GB 13000** 电子邮件传输格式和附录几个部分。由于目前因特网电子邮件只有 **RFC** 规范,没有相应的国际标准和国家标准,因此在本文本中同等参考了与通用电子邮件传输格式相关的内容。附录中收录了与 **GB 13000** 相关的变体格式 **UTF-7**、**UTF-8** 以及编码方式 **Base-64** 的内容,同时给出了基于 **GB 13000** 中文传输的实现参考模型。

本标准由中华人民共和国信息产业部提出。

本标准由全国信息技术标准化技术委员会归口。

本标准起草单位:中国电子技术标准化研究所、北京四通新世纪信息技术有限公司、信息产业部数据通信技术研究所、中国科学院软件研究所、清华大学。

本标准主要起草人:李韵琴、胡万进、陈淑仪、吴健、冯吉祥、陶文星、陈壮。

中华人民共和国国家标准

信息技术 因特网中文规范 电子邮件传送格式

GB/T 18304—2001

Information technology—Chinese specification of
internet character transfer format for email

1 范围

本标准规定了在因特网电子邮件中使用的中文字符信息和其他文种字符的信息传送格式。本标准适用于因特网电子邮件系统以及相关的应用领域。

2 引用标准

下列标准所包含的条文,通过在本标准中引用而构成为本标准的条文。本标准出版时,所示版本均为有效。所有标准都会被修订,使用本标准的各方应探讨使用下列标准最新版本的可能性。

GB/T 1988—1998 信息技术 信息交换用七位编码字符集(eqv ISO 646:1991)

GB 2312—1980 信息交换用汉字编码字符集 基本集

GB 13000.1—1993 信息技术 通用多八位编码字符集 第一部分:体系结构与基本多文种平面
(idt ISO/IEC 10646-1:1993)

GB/T 15273.1—1994 信息处理 八位单字节编码图形字符集 第一部分:拉丁字母一
(idt ISO 8859-1:1987)

RFC 821 因特网文本消息格式标准(Standard for the Format of ARPA Internet text messages)

RFC 822 简单邮件传送协议(Simple Mail Transfer Protocol)

RFC 2045 多用途因特网邮件扩展 第1部分(Multipurpose Internet Mail Extensions Part 1)

RFC 2046 多用途因特网邮件扩展 第2部分(Multipurpose Internet Mail Extensions Part 2)

RFC 2047 多用途因特网邮件扩展 第3部分(Multipurpose Internet Mail Extensions Part 3)

RFC 2048 多用途因特网邮件扩展 第4部分(Multipurpose Internet Mail Extensions Part 4)

RFC 2049 多用途因特网邮件扩展 第5部分(Multipurpose Internet Mail Extensions Part 5)

RFC 1641 在多用途因特网邮件扩展中使用 Unicode(Using Unicode with MIME)

3 定义

本标准采用下列定义。

3.1 纯文本 plain text

由字符、换行控制符和换页控制符等组成的文本。它不提供也不允许格式化命令、字体属性指示、处理指令、解释指令或内容编排等控制功能。

3.2 消息 message

由信封和内容组成的能够在网络上传送的一个信息单位。信封包含提供传送和投递的必要信息,内容由要投递给接收方的对象组成。所涉及的部分只包括消息内容的格式和某些语义以及信封的非技术

国家质量技术监督局 2001-01-02 批准

2001-08-01 实施

规范信息。消息由消息头字段(信封)和正文(内容,可选)组成。正文与消息头之间通过一个空行分开。

3.3 字段 field

在没有特指的情况下,字段是指消息中的有逻辑字符串区域。一个字段可以分为多个子字段,每个子字段也可以是一个有逻辑字符串。

3.4 消息头 header

在没有特指的时候,消息头是指电子邮件消息的信封部分。每一个消息头字段可以看成是一个单独的有逻辑的字符串,包括字段名和字段内容。为了方便,字段内容部分可以分成多行表示,称为“折叠”。消息头中主要有完成传送、投递和扩展的字符类型以及编码等信息。

3.5 正文 body

在没有特指的时候,正文指消息的内容部分。正文是要投递给对方的对象,包括文本、图像、音频、视频和应用程序等。

3.6 字符集 character set

在 **MIME** 里所用的字符集是指一种将八位二进制字节串转换为可识别的字符串的方法。注意,这里并不需要在其他方面进行绝对明确的转换,即不是所有的字符都可以通过给定的字符集进行表述,以及一个字符集可能会提供不止一种将一个八位二进制字节串表示成一个特定字符串的方法。它不仅包括将单个字符直接映射成单个八位二进制字节的规则(如 **GB/T 1988** 和 **GB/T 15273.1**),而且在 **MIME** 里还包括多字节编码字符集和交换技术等。

3.7 多用途因特网电子邮件扩展 MIME

MIME,即多用途因特网电子邮件扩展(**Multipurpose Internet Mail Extensions**),是对 **RFC 822** 以及 **RFC 934** 和 **RFC 2049** 所定义的因特网文本消息格式标准进行的扩展。它对消息格式进行了重新定义,允许:(1)用字符集而不只是 **GB/T 1988** 来表示的文本形式的消息内容;(2)有很多不同格式的可扩展的非文本形式的消息内容;(3)多组分消息内容;(4)用字符集而不只是 **GB/T 1988** 来表示的文本形式的消息头信息。与 **MIME** 有关的是 **RFC 2045**、**RFC 2046**、**RFC 2047**、**RFC 2048** 和 **RFC 2049**。

3.8 charset

charset 是电子邮件 **MIME** 消息头的 **Content-Type** 字段里用来指定“**text/plain**”数据类型的关键参数。它给出了消息中所使用的字符集名称。

3.9 中文字符 Chinese character

在 **GB 13000.1—1993** 中收录并编码的所有汉字、汉语符号、少数民族文字及符号。

4 总要求

4.1 电子邮件格式

4.1.1 因特网文本消息格式标准

因特网文本消息格式标准(**Standard for the format of ARPA Internet text messages**)等同 **RFC 822**。

简单邮件传送协议(**Simple Mail Transfer Protocol**)等同 **RFC-821**。

4.1.2 多用途因特网邮件扩展格式

本部分引用 **RFC 2045** 中与非 **GB/T 1988** 代码及其传送有关的内容,其他部分等同参考 **RFC 2045**、**RFC 2046**、**RFC 2047**、**RFC 2048** 和 **RFC 2049**。

RFC 822 只定义了因特网上 **GB/T 1988** 代码邮件消息的标准传送格式。传送非 **GB/T 1988** 代码文本(其中包括中文)或其他多媒体数据,应采用多用途因特网邮件扩展(**MIME**)对 **RFC 822** 进行扩充所定义的标准格式。这些消息头字段标准格式的正式定义如下:

$$\text{entity-headers} := [\text{content CRLF}]$$

$$[\text{encoding CRLF}]$$

[id CRLF]
 [description CRLF]
 * (MIME-extension-field CRLF)

MIME-message-headers :=entity-headers
 fields
 version CRLF
 ;本定义中所隐含的消息头字段顺序应忽略

MIME-part-headers :=entity-headers
 [fields]
 ;任何不以"content-"开头的字段都可能因没有意义
 ;而忽略,本定义中所隐含的消息头字段顺序应忽略

其各种特定的 **MIME** 消息头字段语法如下:

MIME-Version 消息头字段

用来声明因特网消息正文格式所采用的版本,并且任何按 **RFC 2045** 文档编排的消息都必须包括如下消息头字段:

MIME-Version : 1.0

MIME-Version 字段的正式 **BNF** 定义如下:

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

Content-Type 消息头字段

用来指明在含有 **MIME** 项的正文中数据的性质,给出媒体类型和子类型标识符,以及提供某些媒体类型所需要的辅助信息,以便正在接收的用户代理程序能选择一个合适的代理或机制去向用户表示这些数据,或者用一个合适的方式处理这些数据。**Content-Type** 消息头字段的值称为媒体类型,分为顶层媒体类型(声明数据的通用类型)、子类型(指明与通用类型对应的具体格式)和参数(媒体子类型的修饰部分),其定义如下:

content := "Content-Type" : "type" / "subtype"
 * (" ; "parameter)
 ;媒体类型和子类型的匹配与大小写无关

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 := <由标准途径 **RFC** 定义的并在 **IANA** 注册的扩充标记>

x-token := <字符 "X-" 或 "x-" 后接任何中间没有空格的标记>

subtype := extension-token / iana-token

iana-token := <公开定义的扩充标记。该标记必须按 **RFC 2048** 指定在 **IANA** 注册>

parameter := attribute " =" value

attribute := token

;属性的匹配与大小写无关

value := token / quoted-string

```

token := 1 * <任何除 SPACE, CTLs 和 tspecials 的 GB/T 1988 字符>
tspecials := "(" / ")" / "<" / ">" / "@" /
           ", " / ";" / ":" / "\" / "<" / ">"
           "/" / "[" / "]" / "?" / "="
           ;在参数值里使用时必须以引用串的形式出现

```

这里,

- a) 类型、子类型和参数名均与大小写无关。参数值通常与大小写相关,但有时也有意使用与大小写无关的形式。参数的排列顺序不分先后。
- b) 参数依赖于媒体类型和子类型,是一个可选项,大多数参数都与一个具体的子类型相关。**MIME** 实现程序可以忽略任何无法识别名称的参数。
- c) 初始的五种表示单一媒体组分的标准顶层媒体类型为:
 - (1) **text**——文本信息 子类型“**plain**”用来特指不包含任何格式化命令和指令的纯文本。
 - (2) **image**——图像数据 需要用一种显示设备去查看其信息。定义的初始子类型有 **JPEG** 和 **GIF** 等。
 - (3) **audio**——音频数据 需要用一种音频输出设备去“显示”其内容。
 - (4) **video**——视频数据 是移动的图像,需要通过专用的硬件和软件去播放。初始的子类型是有“**mpeg**”等。
 - (5) **application**——应用类数据 典型情况是非中断性二进制数据和能被某个应用程序处理的信息。子类型“**octet-stream**”用于非中断性二进制数据的情况,“**PostScript**”子类型用于传送 **PostScript** 材料。

初始的两种表示多种组合媒体的标准顶层媒体类型为:

- (1) **multipart**——由多种无关数据类型组成的数据。
 - (2) **message**——已封装过的消息。
- d) 在使用“**text**”媒体类型发送纯文本信息时,使用“**charset**”参数可以指明“**text**”子类型正文文本的字符集,尤其是在包括有表示通用纯文本的子类型“**text/plain**”的时候。

有关 **Charset** 参数的说明如下:

Charset 参数是 **Content-Type** 字段里用来指定“**text/plain**”数据类型的关键参数,给出消息中使用的字符集名称,其形式可有如下两种:

```

Content-type: text/plain; charset = charset _ name
Content-type: text/plain; charset = " charset _ name"

```

Charset 参数的值与大小写无关。**Charset** 参数不出现时的默认字符集是 **GB/T 1988**。

任何将来出现的“**text**”子类型标准规范都必须指出是否同样使用“**charset**”参数以及可能要限定的参数。附加的字符集可以通过 **IANA** 登记注册。

Content-Transfer-Encoding 消息头字段

在简单邮件传送协议里,限制邮件消息为 7 位的 **GB/T 1988** 代码数据,每个文本行的长度(包括 **CR**、**LF**)不超过 1 000 个字符。因此,有必要定义一种将各种媒体数据编码成这种 7 位短行格式的机制。这种编码机制在 **MIME** 里由 **Content-Transfer-Encoding** 消息头字段来指明。**Content-Transfer-Encoding** 字段提供两条信息:指明正文采用了何种编码转换方式以及对应地必须采用何种解码操作才能恢复数据成原样;指出编码结果的范围是什么。**Content-Transfer-Encoding** 消息头字段的取值为一个指定编码类型的单个标记,其正式语法格式和编码类型如下:

```

encoding := "Content-Transfer-Encoding" : "mechanism
mechanism := "7bit" / "8bit" / "binary" /

```

"quoted-printable"/"base64"/
ietf-token/x-token

这些值都与大小写无关。编码类型“7bit”要求正文采用7位的文本邮件表示方式。如果 Content-Transfer-Encoding 消息头字段没有出现,那么其默认取值是假定“Content-Transfer-Encoding:7bit”。

如果有必要,实现机制可以定义私有的“Content-Transfer-Encoding”值,但是必须使用“x-token”来指明是非标准状态,例如“Content-Transfer-Encoding:x-my-new-encoding”。

在 Content-Transfer-Encoding 消息字段的赋值中,有一种是“base64”。这种编码方法可是用一种编码形式(不一定要可读)来表示任意字节串。详细内容见提示的附录 A。

4.2 UCS 字符集传送格式

本标准规定以 GB 13000 作为因特网电子邮件中中文字符信息的传送格式。

根据 4.1.2 对因特网电子邮件中有关文本传送格式的扩充定义,本节定义 GB 13000 作为因特网电子邮件中新的基本字符集,用于传送多八位编码的字符。

所涉及内容主要包括 GB 13000 字符集编码格式和 charset 字段名称两部分。

目前 GB 13000 只定义了 BMP 平面中的字符,随着 GB 13000 新版本的颁布,本标准也将作相应的修订和增补。

4.2.1 UCS 字符集编码格式

定义 GB 13000 作为因特网 MIME 允许的基本字符集之一,即在电子邮件消息中使用 GB 13000 (目前为 GB 13000.1)作为 charset 字段的赋值之一。

在因特网 MIME 消息头中,按如下两种形式声明电子邮件文本的字符集为 GB 13000.1:

Content-Type;text/plain;charset=*ucs-charset-name*

Content-Transfer-Encoding:*encoding-type*

或

Content-Type;text/plain;charset="*ucs-charset-name*"

Content-Transfer-Encoding:*encoding-type*

这里只给出了在电子邮件中传送 GB 13000 字符代码的特定头信息段。其中 *ucs-charset-name* 是标识 GB 13000 字符集的 charset 名称,其具体内容在 4.2.2 和 4.2.3 中说明;*encoding-type* 是传送 GB 13000 字符代码的具体编码类型,可以采用 base64 或其他编码方式。有关 base64 编码的内容,参阅本标准提示的附录 A。

4.2.2 charset 赋值规则

所有为 MIME 注册使用的 GB 13000 变体和版本均使用符合以下统一形式的 MIME charset 名称:

charset=*SID-version*[-*variant*]

其中 *SID* 是字符集标准标识号,即国际标准 ISO/IEC 10646(GB 13000)。*version* 是字符集标准的“版本号”,对 ISO/IEC 10646(GB 13000)而言,其构成形式为 *release*[-*form*],*release* 是 ISO/IEC 10646 (GB 13000)的发行号,目前只有“1”(表示 ISO/IEC 10646.1-1993),*form* 是 ISO/IEC 10646 (GB 13000)的编码形式,目前已经颁布的有 UCS2 和 UCS4 两种形式。*variant* 表示字符集标准的变体形式,目前有 UTF7 和 UTF8 两种,当然也可以不采用变体,而直接使用标准的原始编码。方括号内的参数都是可选的。

4.2.3 charset 赋值列表

根据 4.2.2 中的定义,下面列出了 GB 13000.1—1993(ISO/IEC 10646.1—1993)的 charset 参数名称:

ISO/IEC-10646-1-UCS2

ISO/IEC-10646-1-UCS2-UTF7

ISO/IEC-10646-1-UCS2-UTF8

ISO/IEC-10646-1-UCS4

ISO/IEC-10646-1-UCS4-UTF7

ISO/IEC-10646-1-UCS4-UTF8

本标准的 **charset** 参数的赋值列表,将随着信息技术发展,逐步增补。

附 录 A
(提示的附录)
UTF-7 与 UTF-8

由于 ISO/IEC 10646(GB 13000)标准编码空间十分庞大,传统的内码体系几乎无法表示整个 ISO/IEC 10646(GB 13000)字符集,而且,目前绝大多数的软件、硬件体系和输入/输出设备以及因特网的邮件几乎完全基于 ISO/IEC 646(GB/T 1988)。因此,在实现时就必须考虑兼容性、继承性问题。采用 UCS 编码的变体是解决这一问题的途径之一。

本附录列出了 UCS 编码在具体应用实现中的两种变体形式 UTF-7 和 UTF-8 的编码规则,与 UCS 的映射关系及应用情况。

A1 UTF-7

UTF-7 是主要针对目前基于 RFC 822 的因特网电子邮件系统只能支持 7 位 ISO/IEC 646(GB/T 1988)码传送的现状而提出的一个 UCS 编码的变体形式。UTF-7 采用带换档字符的一个或多个 7 位 ISO/IEC 646(GB/T 1988)字节串表示 UCS 字符串,以便在只支持 7 位传送的邮件系统中传送 UCS 字符。

UTF-7 将 UCS 字符集分为三个子集,D 集,O 集和 B 集。

D 集是可直接编码的字符集,包括:字母 A~Z、a~z、数字 0~9 和以下特殊字符(‘+’和‘=’被忽略,见表 A1):

表 A1 D 集字符

字符	GB/T 1988 和 UCS 值(十进制)
!	39
(40
)	41
,	44
-	45
.	46
/	47
:	58
?	63

O 集是可选直接编码的字符集,包括下列字符(“\”和“~”被忽略,见表 A2):

表 A2 O 集字符

字符	GB/T 1988 和 UCS 值(十进制)
!	33
"	34
#	35
\$	36
%	37
&	38
*	42
;	59
<	60
=	61
>	62

表 A2(完)

字符	GB/T 1988 和 UCS 值(十进制)
@	64
[91
]	93
^	94
-	95
/	96
{	123
	124
}	125

GB/T 1988 字符集中的“\”和“~”不包括在内,因为这两个字符在应用中有不同的含义。

B 集是需要按 Base-64 编码规则编码的字符集(Base-64 参见附录 B)。

用 UTF-7 编码字符串表示 UCS 字符的转换规则如下:

规则 1 (直接编码):在上面定义的 D 集中的 UCS 字符按照 GB/T 1988 直接编码、O 集中的 UCS 字符可以选择性的直接编码,同样地,它们中的许多字符禁止在头域中出现,或者不能正确地通过某些邮件网关。

规则 2 (转换编码):任何 UCS 字符序列都可以使用 B 集中的一列字符编码和一个前导变换字符“+”(GB/T 1988 值为 43),字符“+”后的八进制数被解释为 64 基字符,直到出现一个非 64 基字符,包括控制字符如回车和换行。因此,一个 UCS 字符变换序列通常在一行的行尾结束,作为一个例外,如果一个序更以字符“-”(GB/T 1988 值为 45)结束,则这个字符被忽略,其他结尾字符不能忽略正常处理。作为一个特例,“-”作为“+”的编码。

规则 3:制表符(0x09)、回车(0x0d)和换行(0x0a)字符可以用它们的 GB/T 1988 直接表示。注意, MIME 内容转换编码按规则使用这些字符,这些用法不遵守 RFC 822 的限制。

利用这三个规则,一个 UCS 字符变为 1 行平均 22/3 个 7 位的 GB/T 1988 码字符串。例如:字符串“Hi Mon 你好!”,其 UCS 编码为 0048 0069 0020 004D 006F 006D 0020 转换为 UTF-7 串为:Hi Mon +。

A2 UTF-8

UTF-8 是由 X/Open 联合 I18N 工作组提出的一种与 UNIX 系统兼容的文件系统安全的 UCS 转换格式,并在 ISO/IEC 10646.1(GB 13000.1)附录 R 中推荐。

UTF-8 采用 1~6 个八位字节来表示 BMP 中的一个字符。第一个八位字节从左到右有几个“1”就表示这个字符占几个八位字节,第二个八位字节及以后的八位字节,每一个以“1”打头,作为一个八位字节的标识,以“0”区分标识和后面的“有效位”。所有的“有效位”串接起来表示一个字符真正有意义的“位”。表示原 GB/T 1988 128 字符的 UTF 格式为一个八位字节,以“0”为先导。表 A3 是 UTF-8 编码方案。

表 A3 UTF-8 编码方案

有效位	最小值(Hex)	最大值(Hex)	二进制的位序列
7	00000000	0000007F	0xxxxxxx
11	00000080	000007FF	110xxxxx 10xxxxxx
16	00000800	0000FFFF	1110xxxx 10xxxxxx 10xxxxxx
21	00010000	001FFFFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
26	00200000	03FFFFFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
31	04000000	7FFFFFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

其中的 x 位连接起来就组成了 UCS 码。若只表示 BMP(即 UCS-2),则只需三个八位字节。

UTF-8的特点是兼容现有的基于ISO/IEC 646(GB/T 1988)字符的软、硬件平台,以及UCS码的转换算法,是目前使用比较广泛的一种UCS的变体形式。

对于UTF-16,可以用两个UTF-8编码表示。

附录 B
(提示的附录)
Base-64 字符

Base-64 内容传送编码(Content-Transfer-Encoding)是一种用不可读的编码表示任意 8 位序列的编码形式,其编码和解码算法十分简单。

Base-64 的编码算法是将 24 位一组的输入位串转换成 4 个输出的编码字符。编码过程从左到右,24 位输入串被分解为 4 个 6 位的位组,每个位组的值为 0~63,然后,从表 B1 中取出相应的编码字符,放入输出串中。

编码输出流所表示的行不超过 76 个字符。所有断行或出现在表 B1 以外的字符都将被编码软件忽略。在解码时,表 B1 以外的字符、行中断以及其空白都表示传送出错,并给出警告或出错消息。

表 B1 Base-64 字符

值	编码	值	编码	值	编码	值	编码
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	(pad)=	
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

有关 Base-64 的详细内容参见 RFC 2045。

附录 C
(提示的附录)

有关 Unicode 的 charset 赋值列表

考虑到等同于 ISO/IEC 10646/BMP 的工业标准 Unicode 目前有一定的用户,以及 Unicode 已经发布的版本,特列出如下有关 Unicode 的 charset 参数名称的别名:

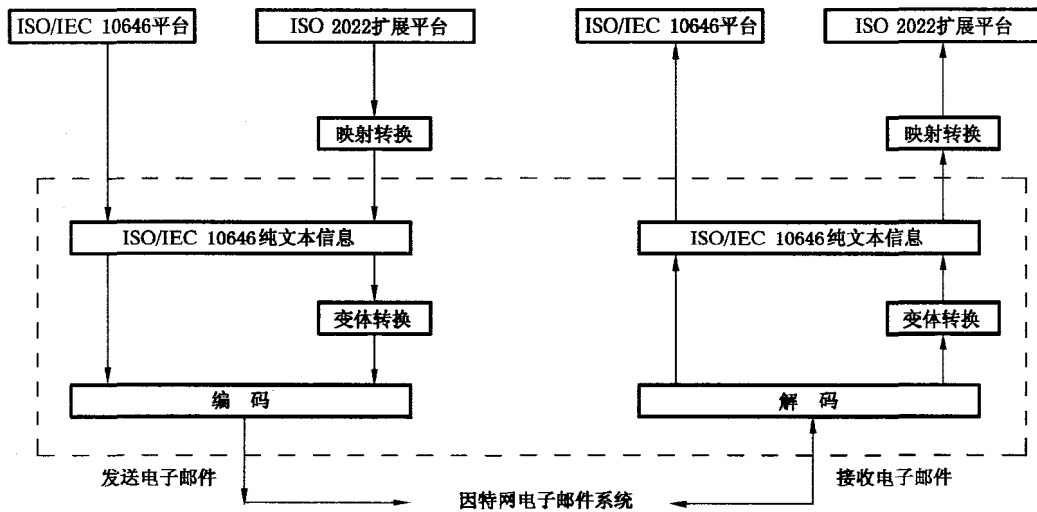
- csUnicode
- Unicode-1-1

Unicode-1-1-UTF-7
csUNICODE11
csUNICODE11UTF7

附录 D
(提示的附录)
实现参考模型

D1 总体结构

本标准所给出的实现参考模型,其总体结构如下图所示。其中,虚线框内的部分为本标准所规定的因特网电子邮件中文消息传送格式。



在电子邮件发送过程中,对于在 ISO 2022(GB 2311)扩展平台上使用的本地字符集纯文本信息,通过映射转换的方法转换为 ISO/IEC 10646(GB 13000)字符集纯文本信息;对于在 ISO/IEC 10646(GB 13000)平台上,直接使用 ISO/IEC 10646(GB 13000)字符集纯文本信息。对于 ISO/IEC 10646(GB 13000)字符集纯文本信息,通过可选的变体转换(如 UTF7, UTF 等),最后对其进行编码,在电子邮件的相应部分中,对所使用的字符集、变体形式和编码类型进行标记,然后向因特网电子邮件系统发送电子邮件。

在电子邮件接收过程中,对于从因特网电子邮件系统接收到的电子邮件,首先按照标记的编码类型,对于其中的纯文本信息进行解码,如果字符集字段有变体转换,进行相应的转换,将其转换为 ISO/IEC 10646(GB 13000)字符集纯文本信息,对于 ISO 2022 扩展平台,通过映射转换,将其转换为本地字符集,以便进行处理。

D2 关键代码段

本标准所给出的实现参考模型的关键代码段采用 C/C++ 伪代码形式,使用 SOCKET 通信接口,分为发送和接收两部分,其中的斜体部分需要按照实际情况作必要的修改。

发送部分:

```
char *SendMessage[] =
    "HELO your_host\r\n",
    "MAIL FROM:your_MailBox@your_host\r\n",
```

```

"RCPT TO:your_MailBox@your_host\r\n",
"DATA\r\n",
"Subject:your mail subject\r\n", //电子邮件开始
"To:your destination address\r\n",
"Mime-Version:1.0\r\n",
"Content-Type:text/plain;charset=ISO/IEC 10646-1-UCS2-UTF7\r\n",
"Content-Transfer-Encoding:7bit\r\n",
...
这里是电子邮件的纯文本信息部分
... //电子邮件结束
"QUIT",
0);

```

SOCKET hsocket;

1. 初始化电子邮件(SendMessage)的相关信息
2. 按照字符集、变体形式、编码类型等,对电子邮件的纯文本部分进行相应的变换和编码,
3. 初始化 SOCKET
4. 建立 SOCKET
5. 连接 SOCKET
6. 发送电子邮件

```

int nIndex=0;
do{
    send(hsocket,SendMessage[nIndex],strlen(SendMessage[nIndex]),NO_FLAG);
    //发送函数
    接收信息,
    如果接收到的信息为出错信息,发送失败,退出循环
    nIndex++;

    }while(SendMessage[nIndex]);

```

7. 关闭 SOCKET

接收部分:

```

char * RecvMessage[] =
"USER your_mailbox\r\n",
"PASS:your_password\r\n",
"STAT\r\n",
"LIST\r\n",
"RETR 1\r\n",
"DELE 1\r\n",
"QUIT",
0);

```

SOCKET hsocket;

1. 初始化 **SOCKET**
2. 建立 **SOCKET**
3. 连接 **SOCKET**
4. 接收电子邮件

int nIndex=0;

do{

send(hsocket,RecvMessage[nIndex],strlen(MailMessage[nIndex]),NO_FLAG);

 //发送函数

接收信息,

如果接收到的信息为出错信息,接收失败,退出循环

nIndex++;

}while(RecvMessage[nIndex]);

5. 关闭 **SOCKET**

对于接收到的电子邮件的纯文本部分,按照电子邮件中所标记的字符集、变体形式和编码类型等,对其进行相应的解码和变换。
