



网络专业人员书库

*Internet Email Protocols:
A Developer's
Guide*

Internet Email 协议 开发指南

(美) Kevin Johnson 著 科欣翻译组 译



机械工业出版社
China Machine Press



Addison-Wesley

网络专业人员书库

Internet Email 协议 开发指南

(美) Kevin Johnson 著

科欣翻译组 译



机械工业出版社
China Machine Press

本书把在许多RFC和软件包中分散的信息集中在一起,描述Internet电子邮件的关键技术,详细介绍Internet电子邮件的基本概念,描述如何应用这些技术提供电子邮件服务的框架。本书还介绍如何发现和解决电子邮件问题,如何评价标准和协议的不同实现,以及如何更好地掌握相关的知识以迎接编写处理电子邮件程序的挑战。它是一本广大电子邮件应用程序开发者的重要参考书。

Kevin Johnson :Internet Email Protocols :A Developer's Guide.

Original edition copyright © 2000 by Addison Wesley Longman, Inc.

Chinese edition published by arrangement With Addison Wesley Longman, Inc. All rights reserved.

本书中文简体字版由美国Addison Wesley 公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号:图字:01-1999-3681

图书在版编目(CIP)数据

Internet Email 协议开发指南/(美)约翰逊(Johnson, K.)著;科欣翻译组译. —北京:机械工业出版社,2000.6

(网络专业人员书库)

书名原文:Internet Email Protocols: A Developer's Guide

ISBN 7-111-08034-3

I. I… II. ①约…②科… III. 电子邮件 IV. TP393.098

中国版本图书馆CIP数据核字(2000)第22084号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:李云静

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2000年6月第1版第1次印刷

787mm×1092mm 1/16·18.5印张

印数:0 001-6 000册

定价:49.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前言

本书介绍构成Internet电子邮件技术的各种协议和标准。描述了电子邮件的格式，包括MIME——SMTP、POP和IMAP协议，过滤、邮件列表技术和安全性。重点是理解Internet电子邮件如何工作和怎样工作，它如何从早期的形式发展成为今天人们日常使用的形式。

有两种类型的应用程序主宰Internet：Web和电子邮件。进入任何书店的计算机图书柜台，数一数专门介绍Web的书籍有多少，包括为终端用户写的书和为开发者和管理员写的书。现在再数一数专门讨论电子邮件的图书有多少，会发现这方面的图书要少得多，特别是关于电子邮件技术方面的图书更少。本书就是为了弥补这方面的不足而编写的。

学习电子邮件如何工作不一定是直观的，这方面的信息涉及分布在多年写的许多文档中，且大多数的协议经历了几代，所以又有更正的版本和新版本。因此理解电子邮件文档与它的前身和后继之间的关系就比较困难。本书通过介绍电子邮件技术的最新状况、描述建立的标准，来说明这方面的信息。在阅读本书之后，读者能更容易理解一个特定的RFC。

另一个困难是大多数标准并不是为了帮助人们学习特定的协议，而是为了提供协议的准确定义，至少是在理论上提供这样的定义。本书按更适合学习这些协议的顺序介绍。

除了这些问题外，在标准和建议的标准的解释方面有差别和错误，在文档本身之中也有二义性和错误，这就产生了以不同方法实现这些标准的各种电子邮件软件包。在本书的某些地方会给出特定问题的主要解释。

最后一个问题是很难得到电子邮件如何工作的全貌，本书的目标是解决这个问题。

本书的读者

本书是为需要了解Internet电子邮件技术的开发者、管理员和超级用户或一般用户编写的。开发者会发现以更易于理解设计的标准文档的形式组织的信息，甚至不创建专门电子邮件应用程序的开发者也会发现对给他们的程序增加电子邮件功能的有用信息。

系统管理员可以使用该信息帮助他们诊断电子邮件系统的问题和管理电子邮件系统。系统管理员在理解了系统的各个组成部分是如何组织在一起和如何工作的时候，生活要轻松得多。超级用户可以使用该信息帮助他们使用电子邮件工具达到极致，特别是介绍信件、MIME、过滤和安全性方面的章节更是如此。

本书的组织

本书的每一章均建立于前一章和前几章的基础之上，初次涉及电子邮件技术方面的读者应从头开始，并逐章学习。高级的读者当然可以按任何顺序阅读。

第2章、第3章、第4章和第9章含有理解电子邮件的基础知识，而其他章节可以根据读者的兴趣跳跃式地阅读。

第1章概述Internet电子邮件。

第2章详细介绍信件的结构。

第3章描述SMTP和对该协议的各种扩展，以及它们是如何用于传送电子邮件的。

第4章描述MIME和其如何用于把结构化的或非文本的数据打包到信件中。

第5章介绍POP协议及其使用和各种扩展。

第6章描述IMAP协议及使用和几个协议扩展。

第7章描述电子邮件过滤器是如何工作的，以及可以使用电子邮件过滤器的各种任务。重点是使用电子邮件如何工作的知识使过滤更有效（特别是处理非请求的大宗电子邮件）。

第8章详细介绍邮件列表处理器及如何操作电子邮件信头执行邮件列表功能。

第9章描述电子邮件安全性的各个方面，包括验证、数字签名、加密和匿名性。

附录A描述几个文件夹格式，重点是它们的各种设计特性的性能含义。

附录B描述与UNIX中文件夹锁有关的各种问题，这些问题都特别麻烦。

附录C介绍几个常见的程序设计语言，以及与电子邮件有关的一些特性。

附录D简单介绍IMSP协议。尽管这个协议没有被广泛实现，而且正被ACAP代替，但也应有所了解。

附录E简单介绍ACAP协议。尽管它不完全是一个电子邮件协议，但却提供了几个电子邮件开发者感兴趣的特性。

附录F简单介绍LDAP协议，提供目录服务的基于网络的查询和更新。

约定

如果一个例子涉及客户机和服务器之间或用户和计算机之间的对话，客户机或用户的输入设置为粗体。对话中需要特别注意的项目以粗斜体显示，信息的汇总以斜体表示。

the client said this

the server responded with *this*

followed by a large amount of other information

大多数电子邮件协议使用一个回车（CR、US-ASCII 13、“\r”）字符后跟一个换行（LF、US-ASCII 10、“\n”）字符终止文本行。为了增加可读性，本书中的例子没有给出行终止符（只有几个例子除外）。每个协议有特定的行终止要求，每个要求应被看成是例子和语法描述中的隐含意义。

在介绍语法描述时，应注意一些语法与RFC中提供的语法不完全匹配，特别是一些老的RFC，如RFC822。为了清楚地表达，作者提供了更容易理解或解决二义性问题的语法描述，特别是RFC822中的空白使用。差别通常是表面的，但表现的“改进”语法描述不要作为参考资料，而是作为辅助的学习资料。

许多例子使用域名example.com、example.edu和example.net。不要向这些地址发送电子邮件，这样做是不行的。在需要域名的位置使用example域，是为了避免使用在将来有人可能会使用的域名。

最后要说明的是，本书在引用电子邮件时使用email，一些人可能觉得应该使用e-mail，但英语有许多字组合、不断被缩写。记住，一开始电子邮件缩写为e-mail。在这种类型的字被

广泛使用时，渐渐地去掉了连字符。事实上，在许多地方简单地使用mail^①。我认为没有连字符的形式更普通。其实，这种形式已经成为电子邮件界标准^②。

关于标准

描述实现Internet电子邮件协议规则和指南的主要文档是RFC，或request for comment。事实上，RFC定义Internet上使用的大多数基本协议。有几千个RFC，从信息到要求的标准。花时间学习Internet协议的任何人最终总会找到需要的信息，因为它们是Internet技术基础的重要方面。第1.2节将进一步描述RFC的创建。

本书要引用大量的RFC，其中的大部分与Internet 直接有关，一些只是间接有关。建议阅读这些RFC，因为这对读者很有帮助。

本书包括电子邮件许多元素的语法描述，在有些地方，这些描述与RFC语法规则中的用词稍微不同，但这是为了更清楚地表达。如果本书中的任何语句与标准冲突，请以标准为准。

最后要指出的是，因为标准和协议是不断发展的，所以对标准和协议的更新作为新的RFC发表。研究Internet标准的读者需要了解特定RFC的历史背景，如它更新或替换哪些RFC，它被哪些RFC更新或替换。本书引用了许多老的RFC，特别是在新RFC的上下文中。毫无疑问，当前的RFC将被新的RFC代替，而另一些仍然是新的。读者应不断检查，看一个RFC是否已经被更新。

得到标准

有两个Internet协议规范的主要来源，一个用于RFC，另一个用于Internet草案。RFC的网址是<http://www.ietf.org/rfc.html>

这个网站有该RFC库和标准过程的概述、对RFC常问问题的回答和RFC文档的实际库。书中引用的所有RFC都可在本书配套光盘上找到。

Internet草案（ID）最终要成为RFC的文档，但在成为RFC前，这些Internet草案要经过几次修改和许多人的审阅。Internet草案库的网址是<http://www.ietf.org/ID.html>

由于Internet草案是变化的，因此在本书配套光盘上没有提供这些草案。这不会是一个问题，因为大多数读者不需要考虑这些草案。

本书配套光盘

本书有一个配套光盘，它含有书中引用的所有RFC，以及几个电子邮件软件包的源代码。还提供了一个HTML层来帮助浏览该信息。这个光盘的目的是允许读者访问这个信息，而不必从Internet下载。

每个RFC以它的发表形式给出，直接来自其来源，没有任何修改。有一个相应的HTML页，显示RFC库主目录中含有一些信息。访问这个光盘上的顶层HTML页可以得到这个信息。

本光盘上的软件包是在本书出版时比较新的版本，Internet上的大多数软件包都是以压缩

① 或许在简写成mail后，每个人还会把它缩写为“m”。

② 关于第二种意见，参见<http://www-cs-staff.stanford.edu/~knuth/email.html>。

的形式发布的，这里的软件包已经被解压缩，并从档案格式扩展，没有任何修改。注意，它们是作为读者查看的例子提供的，作者和出版商都不提供对这些软件包的支持。

最后要说明的是，随着时间的推移，光盘上的一些信息会过时。建议读者检查它们的原始来源，确定其是否是最新的信息。

目 录

前言	
第1章 概述	1
1.1 简况	1
1.1.1 邮件传送代理	2
1.1.2 邮件投递代理	3
1.1.3 邮件用户代理	3
1.2 标准	3
1.2.1 标准的进展	4
1.2.2 关于Backus-Naur范式	5
1.3 Internet电子邮件简史	7
1.4 小结	8
第2章 信件	10
2.1 信件结构	10
2.1.1 信件体	11
2.1.2 信头	12
2.2 字段的一些元素	14
2.2.1 空白	14
2.2.2 原子	15
2.2.3 注解	15
2.2.4 加引号的字符串	16
2.2.5 字段折叠	16
2.2.6 大小写区分	17
2.2.7 退格	18
2.3 标准字段	18
2.3.1 发信方字段	18
2.3.2 收信人字段	19
2.3.3 参考字段	20
2.3.4 跟踪字段	22
2.3.5 重发字段	23
2.3.6 其他字段	24
2.3.7 扩展字段	25
2.3.8 非标准字段	26
2.4 一些例子	27
2.4.1 简单信件	27
2.4.2 重新分发	28
2.4.3 转发	28
2.4.4 回复	30
2.5 电子邮件地址	33
2.5.1 信箱	33
2.5.2 组列表	36
2.6 Received字段	36
2.6.1 字段组成	36
2.6.2 注解	38
2.6.3 机器分析	38
2.6.4 诊断问题	38
2.7 Date-time字段	40
2.8 小结	42
第3章 邮件传送	43
3.1 接口模型	43
3.1.1 命令和应答	45
3.1.2 SMTP会话	48
3.1.3 协议状态	52
3.2 标准命令	53
3.2.1 退出	53
3.2.2 帮助	53
3.2.3 空操作	54
3.2.4 验证	54
3.2.5 扩展	55
3.2.6 问候	56
3.2.7 邮件来自	58
3.2.8 收信人	59
3.2.9 数据	60
3.2.10 复位	61
3.2.11 发送	61
3.2.12 Send或Mail	61
3.2.13 Send和Mail	62
3.2.14 对调	62
3.2.15 本地扩展命令	62

3.3 邮件事务	62	4.4.1 text	95
3.3.1 邮件发送	62	4.4.2 Image、Audio和Video	99
3.3.2 中继	64	4.4.3 Application	99
3.3.3 转发	65	4.4.4 信件	101
3.4 操作考虑	65	4.4.5 Multipart	111
3.4.1 协议不一致	65	4.4.6 其他媒体类型	116
3.4.2 可靠性和责任	66	4.5 信头扩展	117
3.4.3 排队	66	4.5.1 字符集	117
3.4.4 发送	66	4.5.2 编码	117
3.4.5 接收	67	4.5.3 编码的使用	118
3.4.6 超时	67	4.6 mailcap文件	119
3.4.7 弹回	67	4.6.1 一般语法	119
3.4.8 环路检测	69	4.6.2 字段	120
3.4.9 安全	70	4.6.3 标志	122
3.4.10 信件提交	70	4.7 小结	122
3.5 SMTP扩展	71	第5章 POP	124
3.5.1 信件尺寸声明	72	5.1 接口模型	124
3.5.2 命令流水	73	5.1.1 命令和应答	126
3.5.3 8位MIME传送	74	5.1.2 协议状态	127
3.5.4 远程信件排队开始	74	5.2 标准命令	128
3.5.5 增强的状态代码	75	5.2.1 用户	128
3.5.6 投递状态通知	77	5.2.2 密码	129
3.5.7 LMTP	80	5.2.3 APOP	129
3.5.8 提交	81	5.2.4 退出	130
3.6 小结	82	5.2.5 空操作	131
第4章 MIME	83	5.2.6 状态	131
4.1 MIME的先驱者	83	5.2.7 列表	131
4.2 MIME概述	84	5.2.8 取出信件	132
4.3 MIME信头	87	5.2.9 前几行	133
4.3.1 MIME-Version	87	5.2.10 删除	134
4.3.2 Content-Type	87	5.2.11 复位	134
4.3.3 Content-Transfer-Encoding	88	5.2.12 惟一ID列表	134
4.3.4 Content-ID	92	5.3 POP对话举例	135
4.3.5 Content-Description	92	5.3.1 基本对话	135
4.3.6 Content-Disposition	92	5.3.2 信箱信息汇总	136
4.3.7 Content-MD5	93	5.3.3 在服务器上保持最近的信件	137
4.3.8 Content-Language	94	5.4 POP扩展	138
4.3.9 MIME扩展字段	95	5.4.1 扩展服务功能	138
4.4 MIME媒体类型	95	5.4.2 POP3扩展机制	141

5.5 小结	143	6.5.16 lsub	173
第6章 IMAP	144	6.5.17 copy	173
6.1 接口模型	144	6.5.18 append	173
6.2 协议状态	146	6.5.19 check	175
6.3 命令和应答	147	6.5.20 search	175
6.3.1 状态应答	148	6.5.21 fetch	178
6.3.2 服务器数据	151	6.5.22 store	184
6.3.3 命令继续请求应答	152	6.5.23 expunge	185
6.3.4 关于多个等待处理的命令	152	6.5.24 uid	185
6.4 数据对象	155	6.5.25 试验命令	186
6.4.1 NIL	155	6.6 IMAP对话举例	186
6.4.2 原子	155	6.6.1 替换POP	186
6.4.3 加引号字符串	155	6.6.2 信箱信息汇总	188
6.4.4 文字量	155	6.7 IMAP扩展	189
6.4.5 圆括号列表	156	6.7.1 非同步文字量	189
6.4.6 信箱名	157	6.7.2 Idle	189
6.4.7 内部日期	159	6.7.3 名字空间	190
6.4.8 大小	159	6.7.4 访问控制列表	191
6.4.9 信件号	159	6.8 小结	193
6.4.10 惟一标识符	160	第7章 过滤	194
6.4.11 信件标志	160	7.1 过滤的模型	194
6.4.12 信封结构	161	7.1.1 可用的输入	195
6.4.13 信件体结构	163	7.1.2 条件匹配	198
6.5 标准命令	165	7.1.3 过滤器动作	199
6.5.1 capability	165	7.2 邮件代理中的过滤	200
6.5.2 noop	166	7.2.1 MTA过滤	201
6.5.3 logout	167	7.2.2 MDA过滤	203
6.5.4 login	167	7.2.3 MUA过滤	204
6.5.5 authenticate	167	7.3 基本过滤	207
6.5.6 list	168	7.3.1 简单的个人过滤	207
6.5.7 create	169	7.3.2 度假处理	207
6.5.8 delete	169	7.3.3 电子邮件机器人	210
6.5.9 rename	170	7.4 非请求的电子邮件	210
6.5.10 status	171	7.4.1 中继	211
6.5.11 select	171	7.4.2 已知的“犯罪者”	212
6.5.12 examine	172	7.4.3 发信人和收信人字段	213
6.5.13 close	172	7.4.4 字段内容	215
6.5.14 subscribe	172	7.4.5 暴露内情的字段	215
6.5.15 unsubscribe	173	7.4.6 信头错误	215

7.4.7 信件体	216	9.2.2 密码散列函数	243
7.4.8 可选删除列表	217	9.2.3 对称加密	243
7.4.9 白名单	217	9.2.4 非对称加密	244
7.4.10 预防	217	9.2.5 ElGamal	246
7.5 高级过滤	218	9.3 安全服务	246
7.5.1 计分法	218	9.3.1 CRAM	246
7.5.2 协作过滤法	218	9.3.2 OTP	246
7.5.3 动态电子邮件地址	219	9.3.3 Kerberos	249
7.6 小结	219	9.3.4 GSS-API	250
第8章 邮件列表处理	220	9.3.5 OpenPGP	250
8.1 邮件列表的先驱者	220	9.4 安全框架	253
8.1.1 个人地址列表	220	9.4.1 TLS	253
8.1.2 组列表	221	9.4.2 SASL	255
8.1.3 MTA别名列表	221	9.5 协议安全扩展	258
8.2 基本列表处理	222	9.5.1 IMAP扩展	259
8.3 高级列表处理	228	9.5.2 POP扩展	259
8.3.1 弹回处理	228	9.5.3 SMTP扩展	259
8.3.2 管理命令	231	9.6 MIME安全	262
8.3.3 摘要	233	9.6.1 安全多部件	262
8.3.4 归档	234	9.6.2 PGP	264
8.3.5 分布式列表处理器	235	9.6.3 S/MIME	266
8.4 小结	236	9.7 匿名邮件中转站	266
第9章 安全性	237	9.8 小结	267
9.1 安全问题	238	附录A 文件夹格式举例	269
9.1.1 窃听	238	附录B UNIX文件夹锁定	274
9.1.2 假冒	239	附录C 程序设计语言	278
9.1.3 拒绝服务攻击	239	附录D IMSP	281
9.1.4 系统完整性	240	附录E ACAP	283
9.2 核心安全技术	242	附录F LDAP	285
9.2.1 伪随机数的产生	242		

第1章 概 述

电子邮件是计算机用户用来相互发送信件的一组机制。一个用户写好一封信，提供一个或多个收信人地址，然后发送。这封信最终到达它的目的地，收信人使用一个程序就可以看到信的内容。尽管这个过程从概念上讲很简单，但它涉及相当复杂的一组协议、标准和约定。

Internet电子邮件由一组RFC(Request for Comment)文档定义，这些文档由IETF (Internet Engineering Task Force) 出版。它一开始是在Internet的早期设计的，那时大多数用户是Internet研究和开发群体的一部分。随着Internet由越来越多的大学、企业和用户访问，Internet电子邮件也得到很大的发展，它已经成为今天使用的更常用的邮件系统之一。

电子邮件不仅是Internet上最早大量使用的应用程序之一，而且有越来越发展壮大的趋势。几个新的应用程序协议正逐渐替换老的协议，或者使这些老协议越发不被利用。例如WWW (World Wide Web)。老的FTP (文件传输协议) 仍广泛使用，但WWW已经不怎么使用FTP进行文件传输。另一方面，传统的Internet电子邮件仍在不断壮大，而且丝毫没有被替代的迹象。

尽管Internet电子邮件被广泛使用，但它不是唯一的电子邮件类型。现在有几种电子邮件在使用，每一种都在不同的环境中创建，有不同的目标。本书不讨论这些电子邮件，但了解它们并没有坏处。经常出现与这些类型的电子邮件接口的问题。

几年前，许多个人计算机用户经历的第一次电子邮件“爆炸”是通过基于LAN的电子邮件软件包、大型计算机系统或闭合的系统联机服务实现的，它们常常是专有的，往往不能与其他的电子邮件系统互操作，而且可扩展性不好。

为了创建一个世界范围的电子邮件系统，ISO (国际标准化组织) 和ITU (国际电信联盟) 发布了一组信件传递标准，叫做X.400。实际上，这是简写的名字，真正的名字是ISO/IEC 10021，尽管很少叫这个名字。这是一个内容非常丰富、功能非常强且数量大的标准集合。尽管X.400引起了一些人和机构的兴趣，特别是在欧洲，但由于它的规模和复杂性，因此并没有得到Internet 的广泛接受。

Internet的爆炸性增长和流行使大多数其他的电子邮件实现要么采用Internet标准，要么被使用Internet标准的电子邮件系统替换。

最后要说明的是，本书主要讨论Internet电子邮件，但为了简单起见，本书后面将使用电子邮件或邮件这样的术语，而不使用Internet电子邮件，除非要进行非Internet电子邮件的直接比较。

1.1 简况

那么，电子邮件是什么样子，它又是如何操作的呢？在最低的层次，它会很复杂，而在最高的层次，它又很简单。它由信件、把信件从一个地方传递到另一个地方的协议和用户执行各种相关任务的接口组成。

电子邮件的主要部分是信件(message)，大多数的协议和规范都是围绕描述信件是什么样子的，或者如何把信件从一个地方传递到另一个地方为中心的。信件就是媒介。

尽管信件要在第2章详细说明,但在此首先介绍一些有关它的基本知识。信件由信封、信头和信件体组成。信封指定信件的来源和目的地,这将在第3章详细介绍;信头含有关于该信件的各种信息;信件体含有发信人希望传递的信息。在这3个组成部分中,大多数用户只能看到信头和信件体,信封通常只在内部使用。信头和信件体的总体结构是相当简单的,如下面的例子所示。

```
From: bob@flugelhorn.example.com
To: joe@tuba.example.com
Subject: An ineffable email test
Date: Sat, 1 Apr 2000 12:00:00 -0700
```

```
Hey Joe,
Why do you have flour in your hair?
```

```
--
thx,
Bob
```

信头和信件体由一个空行分开。这个例子在信头中有4个字段: To、From、Subject和Date。在信头中可能还有许多其他的字段。

信件一个非常重要的方面是地址,在前面的例子中有两个地址。地址在信封和信头中使用,与信件要发送到哪里和信件来自何方有关。

尽管电子邮件地址可以相当复杂,但基本结构是很简单的:

```
bob@flugelhorn.example.com
```

左边标识邮件要投递到的信箱,而右边标识信箱所在的机器;@符号分开这两个部分。实际上,这个描述掩盖了一些细节,但现在这样的描述就足够了,后面将给出更详细的描述。传统的信件体也是非常简单的,由若干行正文组成。尽管有一些扩展,对信件施加一些结构上的定义,但它通常是自由体的文字数据。对于普通文字这足够了,但因为电子邮件如此广泛地作为通信媒体,所以人们最终希望发送非普通文字的数据。在早期这不是太大的问题,人们把现有的一些技术应用在一起就可以了,但随着时间的推移,明显地需要一些更可靠的技术,所以产生了MIME(多用途Internet邮件扩展),这些扩展定义了一种手段,把不是面向文字的数据打包到信件中。第4章将详细描述这些扩展。

信件自身的用途是有限的,需要一种把它从一个地方传递到另一个地方的方法。这个工作分成几个任务。一个MTA(邮件传送代理)路由邮件,一个MDA(邮件投递代理)投递邮件,一个MUA(邮件用户代理)为用户提供一个界面。这些任务通常由不同的程序完成。尽管有些电子邮件程序可以执行这其中的多个功能,但最好还是把它们分开。

1.1.1 邮件传送代理

MTA从各种来源接收邮件。在收到每个邮件时,MTA就确定这个邮件要路由到哪里和如何路由,如果必要的话,它重新写地址,然后把邮件交给MDA投递。

控制Internet邮件路由的主要机制是DNS(域名服务)协议。DNS提供一个分布式的数据库,把域名映射到几个类型的信息,包括邮件路由指令。大多数的MTA还提供它们自己的机制直接控制路由,作为DNS路由的补充。这对于本地的技巧和解决临时的DNS路由问题是有用的。

在某些情况下，在把邮件从一个地方传送到另一个地方时，需要重写地址。例如，一些机构希望邮件地址表现为来自一个中心位置。重写是这样做的一种方法。

在电子邮件路由进出非Internet电子邮件系统的情况下，也需要重写地址。例如，UUCP (UNIX to UNIX Copy) 使用“!”分割一个特定邮件路由经过的机器列表。因为这个格式与Internet电子邮件的格式不同，路由邮件经过这两个系统的MTA时需要能够相应地重写地址。大多数（如果不是全部）MTA提供一种方法更改重写规则，一些把这些规则编译成程序，而另一些把这些规则放到运行时配置文件中。

1.1.2 邮件投递代理

一旦MTA收到一个邮件，处理这个邮件，并确定了它的路由，就把它交给MDA。MDA负责把该邮件投递到另一个位置，这个位置可以是另一个MDA、一个用户的信箱或执行特殊任务的程序。根据这次投递是成功还是产生永久或临时的故障，MDA决定这次事务是完成、产生一个错误返回给发信人，还是在将来重新发送。

最简单类型的MDA是一些系统用于投递到本地信箱的MDA，它简单地把到达信件放到本地用户的收件箱。但是，在投递时，可以对信件做一些其他的事情。一些MDA不是简单地附加到达的电子邮件，而是提供过滤特性，对到达的信件提供额外的操作。其他MDA可以把电子邮件传递到另一个机器。这些可能性似乎是无止境的。第7章讨论MDA的一个更常见的使用——过滤功能。

如果一个MTA决定信件需要路由到另一个MDA，它把信件提交给一个使用SMTP（简单邮件传输协议）的MDA，这个协议定义一组把信件传递到远程MDA的命令。这个MDA常常在MTA中创建。第3章描述SMTP和它与MTA及邮件投递的关系。

1.1.3 邮件用户代理

MTA和MDA负责信件的路由和传送，而邮件用户代理（MUA）负责为用户提供管理邮件的界面。这个管理通常包括查看信件、管理邮件夹、写和发送新信件，以及回复信件和把现有的信件发送给其他用户。它通常是与信件直接打交道的唯一程序。

在早期的电子邮件中，MUA通常在用户接收电子邮件的相同机器上，最终创建了两个协议：POP（邮局协议）和IMAP（Internet信件访问协议），允许使用MUA阅读位于远程机器上的电子邮件。

POP为MUA提供一个协议，从远程服务器下载用户的收件箱，允许用户使用不总是连接到网络的机器，这在第5章中描述。

IMAP为MUA提供一个协议，操作远程服务器上的邮件文件夹。实现该协议的MUA可以连接到远程IMAP服务器，并执行对信箱和信件需要做的各种任务。这允许用户的邮件文件夹和MUA放在不同的机器上。第6章详细描述这个协议。

1.2 标准

与其他Internet应用和协议一样，标准定义邮件如何操作，这些标准在一组RFC中定义，它们定义使电子邮件经过Internet所必须的协议和格式。为了理解这些RFC的创建和发展，需要理解这些标准发展的知识。

1.2.1 标准的进展

在树的顶部是ISOC (Internet Society, Internet学会), 这是一个国际化的组织, 协调和维护Internet的发展。IAB (Internet Architecture Board, Internet体系结构理事会) 是一个ISOC管理的技术咨询组, 定义Internet的总体结构。除了其他的一些事情外, 它主要提供对标准进展的观察, 并作为Internet RFC文档的出版者。

IETF是IAB管理的实体, 设计和开发Internet上使用的标准和协议。它分成一些领域, 这些领域是一些集中于特定标准工作的工作组。目前的领域包括应用、一般、Internet、操作与管理、路由、安全、传送和用户服务。与电子邮件有关的工作在应用领域中实现。

一个领域中的工作组集中于标准发展的不同主题, 它们根据特定主题的出现和消失而生灭。有时候在创建一个工作组之前为特定的一个主题形成一个BOF, BOF的目的是分析感兴趣的特定领域, 确定工作的需要, 甚至可能创建一些基本的Internet草案来推动标准的发展。如果这个主题是值得的, 则BOF发展为一个工作组。

当一个工作组是活动的, 就会产生一个或多个Internet草案, 最终的产品通常是一个或多个RFC。一个Internet草案的中间文档用于交流该工作组完成的工作。这些草案由IESG (Internet Engineering Steering Group, Internet工程转向组) 审查, 它决定一个Internet草案是否提升到RFC状态。

但在创建一个RFC后, 这个工作还没有完。如果这个规范要成为一个Internet标准, 它应该经过一个标准路线, 评价它的成熟性。成熟性的级别包括:

- 建议标准。这是一个进入级位置, 也就是一个RFC。在标准发展的这个位置, 该规范被考虑是相对稳定的, 得到很好地理解, 进行了很好地审查, 对Internet界有足够的兴趣。下一级的主要障碍是该规范需要更多的宣传, 必须有一些实现确认提到的一些特征。

建议标准应看成是稳定但不成熟的, 规范的内容可能随早期实现的结果而改变, 所以对这种协议的使用要小心为好。

- 草案标准。这个成熟级代表一个规范的重要状态进展, 它意味着自己是成熟和有用的。在这个阶段, 该规范有了几个实现, 并由受影响的团体评价, 对它进行了任何必要的调整。

实现者通常把一个草案标准作为最终规范, 任何进一步的变化可能仅解决特定的问题, 所以在生产环境中使用草案标准的实现也是合理的。

- Internet标准。到达这个成熟级别意味着该规范已经得到很大成功, 并获得足够的经验。在规范中表现高层次的技术成熟性, 通常认为对Internet有很大的好处。

除了成熟级别外, 每个标准文档还被指定一个协议状态, 反映要求该协议的程度:

- 必需的——一个系统必需实现该协议。
- 建议的——一个系统应该实现该协议。
- 选择性的——一个系统可以决定不实现该协议。
- 限制使用——该协议在限制的环境中使用, 通常因为它是实验性的或者是过去历史的协议。
- 不建议使用——一般情况下建议不要使用该协议, 通常是因为功能有限或者是实验性的或阶段性的。

如果一个RFC不打算成为Internet标准，或者还没有进入标准发展轨道，它应标为下面的一个类型：

- 实验性的——通常用于研究和开发工作。
- 信息性的——通常用于把一般信息与Internet团体关联的RFC。它们不必代表一个联盟，也不打算作为一个建议。
- 当前最好的实践——通常用于目前在Internet上最好实践的RFC。RFC1818（当前最好实践）描述这个分类。
- 历史的——通常用于已经被新的RFC代替，或者由于某个原因而认为过时的RFC。

一些RFC含有对唯一名称、号码和参数的要求。多年来，由美国政府建立的IANA（Internet Assigned Numbers Authority, Internet编号管理局）提供这类信息的协调、分配和注册服务。但是，因为Internet是一个国际性的实体，注定需要一个国际化的机构来提供这些服务。这个机构是非盈利公司ICANN（Internet名称和号码分配公司）。

本节提供了Internet标准发展的非常高层次的概述，没有说明一些非常细节的内容。关于维护Internet的一些组织的更多信息，可以参看它们各自的网页。

1.2.2 关于Backus-Naur范式

许多RFC需要指定它们描述信息的语法，一些早期RFC使用BNF（Backus-Naur范式）表示法。最后，发明了一个BNF变种，允许用更紧凑的语法表示。这个ABNF（扩展Backus-Naur范式）用于许多电子邮件RFC中。一开始，许多RFC包括ABNF语法的描述。为了避免信息的重复，其他RFC常常简单地引用含有这个描述的RFC。

最后，创建了一个RFC，把ABNF有关的信息集中到一个文档中，所有RFC都引用这个文档，结果就是RFC2234（扩展BNF语法规范）。

许多电子邮件RFC都主要依赖于ABNF。正因为如此，理解什么是ABNF、它的形式、它是如何工作的就很重要。

一个ABNF语法描述由一组规则组成，这些规则描述一个数据流的结构。每个规则由规则名称、“=”符、规则的说明和行终止符组成。

```
name=elements CRLF
```

规则名称由一个起始字母，后跟字母、数字字符和连字符的任意组合组成。它是不区分大小写的，意味着大写字母和小写字母的任意组合引用相同的规则。

“=”符后面的元素是其他规则名称、操作符、各个字符和字符串的任意组合，这些规则名引用其他的规则。操作符控制这些规则名和值如何组合形成一个指定规则。

单个字符以一个“%”，一个数基指定符和一系列的数字代表。数基可以是表1-1中描述的3个字符之一，所以42的十进制、二进制和十六进制表示分别是“%d42”、“%b101010”和“%h2A”。

字符的区间可以使用这个表示法的变化来指定。

```
%d48-57.
```

这个例子表示包含在字符US-ASCII“0”和“9”之间的闭区间的任何字符。

也可以表示一个序列的字符。在这种情况下，组合两个或多个字符，使用“.”分开它们。下面的例子指定US-ASCII字符“B”、“o”和“b”。

```
%d66.111.98
```

字符串由包含在一对 " 字符内的一些字符表示。

```
"Bob"
```

在ABNF中, 这些加引号的字符串是不区分大小写的, 所以字符串 "bOB" 与上面的例子是一样的。对于不区分大小写的字符串, 必须使用前面的 "." 表示法。

现在, 可以定义一些简单的规则。

```
owner = person
person = "bob"
```

在这个例子中, owner由一个person组成, 一个person由字符串 "bob" 组成。

这看起来好像不是十分有用, 但当增加操作符时就会发挥强大威力。操作符控制规则名、字符和字符串如何组合在一起。

并置(concatenation)操作符允许把一系列的元素组合成一个有序的成员列表。

```
owner = firstname middlename lastname
firstname = "bob"
middlename = "gold"
lastname = "doubloon"
```

在这个例子中, owner定义为firstname后跟lastname的一个实例。

要指定一个替换, 使用 "/" 操作符:

```
pet = dog / cat
```

这个操作符也可以用于组合两个以上的规则。

```
pet = dog / cat / iguana
```

替换操作符的一个变化可以用于给现有的规则增加一个替换。在这种情况下, 使用 "=/" 代替 "="。

```
pet =/ rock
```

使用下面的表示法指定重复。

```
m*n element
```

m和n是可选的十进制值, 指示element至少出现m次, 最多出现n次。

```
1*3pet
```

这个例子指定pet出现1~3次。

如果不指定m, 它缺省为0; 如果不指定n, 它缺省为无穷大。所以, 下面的例子指定3次或更多次的pet。

```
3*pet
```

如果m和n相同, 意味着完全相同的值。在这种情况下, "m*n" 的表示法可以简化为 "n"。

```
2pet
```

该表示法的另一个好处是使用可选序列(optional sequence)。使用 "1*" 指示一个元素的0次或1次出现, 该元素可以放在一对 "[]" 字符中。

```
[element]
```

表1-1 数基指示符

字 符	描 述
b	二进制
d	十进制
x	十六进制