

普遍适用性 (UA) 简介

普遍适用性指导小组 (UASG)

2019/9/23



目录

关于本文件	4
目标受众	4
背景概念	5
域名	5
国家和地区顶级域 (ccTLD)	5
通用顶级域 (gTLD)	5
域名国际化	6
普遍适用性 (UA) 需求	6
U-标签和 A-标签	6
国际化电子邮件地址 (EAI)	7
动态链接生成 (链接化)	8
根区注册管理机构的动态属性	8
普遍适用性行动方案	8
普遍适用性的五大标准	8
用户场景	10
违背国际惯例	12
UA 就绪技术要求	12
整体要求	12
开发人员注意事项	13
设计符合兼容性和灵活性要求的软件	13
通过开发和更新软件实现 UA 就绪的良好实践	14
域名权威性来源: DNS 根区和 IANA 列表	20
包含 IDN 的电子邮件以及为何它不同于 EAI	20
链接化及其挑战	21
良好实践建议	21
统一码 (Unicode) — 背景和码点属性	22
UTF8、UTF16 及其他编码方法	22
IDNA — 简史和现状	23
测试用例	23
升级 EAI 软件	23
高级主题	24
复杂文字	24
从右到左语言和统一码 (Unicode) 一致性	24
Bidi 算法	24



Bidi 域名规则	25
连接符	26
同形异义词和易混淆字	26
规范化、大小写折叠和字符串准备	27
大小写折叠和映射	28
术语表和其他资源	29
术语表	29
RFC 和关键标准	32
关键标准	35
在线资源	36



关于本文件

互联网技术（包括其命名组成）不断发展变化。近年来，互联网名称与数字地址分配机构 (ICANN) 批准了大量新顶级域 (TLD)，一些新 TLD 包含传统 ASCII 字符，另一些新 TLD 则包含非 ASCII 字符（国际化域名）。例如 .nyc、.संगठन、.eco 和 .католик。但是，很多应用程序和服务尚未更新，无法管理规模不断壮大的 TLD。此外，现行互联网电子邮件标准允许在电子邮件地址中使用非 ASCII 字符；因而，除非升级软件，否则将无法妥善处理这些域和地址。这将会对用户体验产生多重影响：

- 无法识别或接受有效电子邮件地址。
- 在浏览器地址栏中，域名被误认为搜索项。

除非软件识别并处理所有域名和电子邮件地址（这种状态称之为普遍适用性 (UA)），否则无法为所有互联网用户提供统一的良好体验。本文件对“普遍适用性”进行了概括性介绍，致力协助开发普遍适用性准备就绪软件。

目标受众

本文档旨在面向技术受众（开发人员、管理人员和运营人员）介绍普遍适用性，此类技术受众可能熟悉某些领域的互联网技术，但不一定详尽了解新 IDN、域名和电子邮件地址对其接受、验证、存储、处理和显示方式的影响。尽管技术背景各异，但所有受众均可以本文件为起点开始了解普遍适用性。



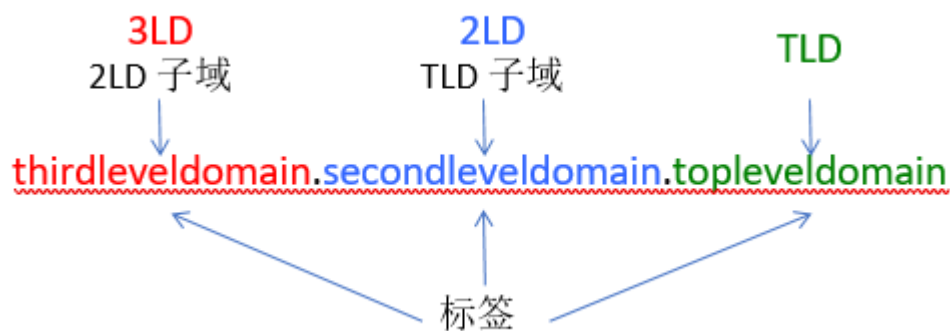
背景概念

域名

域名是计算机和网络在互联网上的人类可读标识符。通常由一串用“点”（句点或句点标点符号）分隔的文本标签表示；例如，`www.example.tld`。每个标签代表域名系统 (DNS) 层次结构的一个级别。

层次结构的最高级别或“根”是顶级域 (TLD) 标签（如 `com`、`jp`、`বাংলা`），显示于域名末尾。由于显示在末尾，人们有时也将 TLD 称为“后缀”。

从根开始深入了解 DNS 层次结构，下一个标签标识的是 TLD 子域，通常称为二级域；再下一个标签标识的是二级域的子域，通常称为三级域；依此类推，每个标签与其相邻标签之间用点分隔。例如，包含三个级别的域名如下所示：



国家和地区顶级域 (ccTLD)

一些 TLD 被授予特定国家或地区。此类域称之为国家和地区顶级域 (ccTLD)。过去，所有 ccTLD 均为两个字母，对应于国际标准化组织分配给国家或地区的 ISO 3166 双字母代码。自 2010 年以来，还广泛采用国际化 ccTLD，使用国家或地区的本土文字表示该国家或地区的名称。

通用顶级域 (gTLD)

大部分非 ccTLD 的 TLD 均称为通用顶级域 (gTLD)，要么面向所有人开放，要么仅限于既定社群成员使用。既包括常见的 `.com`、`.net` 和 `.org`，也包括最近新增的一些域。

通过新 gTLD 项目（一项由 ICANN 协调实施的举措），域名系统 (DNS) 在引入新通用顶级域后呈指数增长。这些新 gTLD 可以代表品牌、兴趣社群、地理区域（城市、地区）等。



域名国际化

起初，域名仅限于 ASCII 字符子集（字母 a-z、数字 0-9 和连字符 "-"）。自 1985 年全球首个 .com 域 symbolics.com 注册以来，域名数量和特征均不断增多，这充分表明全球互联网使用需求日渐高涨，迫切需要将互联网作为公共资源。如今，大部分互联网用户的母语并非英语，但英语仍是互联网主导语言。为帮助推进互联网国际化，互联网工程任务组 (IETF) 自 2003 年开始出台标准，通过翻译机制为部署国际化域名 (IDN) 提供技术指导，支持使用任何统一码 (Unicode) 支持的文字设置域名的非 ASCII 表示（例如，普遍接受-测试.世界、ua-test.كاثوليك 等）。

ICANN 董事会于 2009 年 10 月批准新 IDN ccTLD 引入流程，首批 IDN ccTLD 已于 2010 年 5 月添加到根区。2011 年 6 月，董事会批准并授权启动新 gTLD 项目，其中包括新 ASCII 及 IDN TLD。该项目的首批 TLD 已于 2013 年添加到根区。

普遍适用性 (UA) 需求

迄今据 IETF 发布 IDN 相关指南已达十年之久，目前已有 1000 多个新 TLD 通过 ICANN 新 gTLD 项目投入使用。但是，某些软件 and 应用程序已过时，无法处理这些新 TLD。这会为互联网用户（包括使用非 ASCII 字符和文字的用户）造成很多困扰。普遍适用性可确保所有支持互联网功能的应用程序、设备和系统以统一方式正确接受、验证、存储、处理以及显示全部有效域名和电子邮件地址。例如，每一个有效网址均可解析为正确网站上的预期资源，每一个有效电子邮件地址都能将邮件传送给预期收件人。

普遍适用性指导小组 (UASG) 是在 ICANN 的支持下提出的一项互联网社群倡议，任务是开展相关活动，有效提升普遍适用性并帮助确保为全球互联网用户提供统一的良好体验。

U-标签和 A-标签

使用非 ASCII 字符的域名称为国际化域名 (IDN)。域名的国际化部分可以是任何标签 — 而不仅限于 TLD。

鉴于过去 DNS 本身仅使用 ASCII，因而必需额外创建编码，以便将非 ASCII 统一码 (Unicode) 码点表示为 ASCII 字符串。实现统一码 (Unicode) 转 ASCII 编码的算法称为“国际化域名编码 (Punycode)”算法；输出字符串称为 A-标签。A-标签有别于普通 ASCII 标签，因为 A-标签始终以下列四个字符开头：

xn--

此类字符称为 ACE 前缀。¹

国际化域名编码 (Punycode) 转换可逆：可以从统一码 (Unicode) 转为 A-标签，也可以从 A-标签转回统一码 (Unicode) 字符串（称为 U-标签）。

¹ ASCII 兼容编码 (ACE) 前缀，用于区分国际化域名编码 (Punycode) 编码的标签与其他 ASCII 标签。



国际化域名编码 (Punycode) 算法的唯一标准用途是表达国际化域。假定可以使用国际化域名编码 (Punycode) 编码其他 UTF-8 字符串，但这属于非标准做法且无法与其他系统互用。

(虚构) IDN 示例

U-标签版本	A-标签版本
example.みんな	example.xn--q9jyb4c
大坂.info	xn--uesx7b.info
みんな.大坂	xn--q9jyb4c.xn--uesx7b

国际化电子邮件地址 (EAI)

电子邮件地址包含两部分：

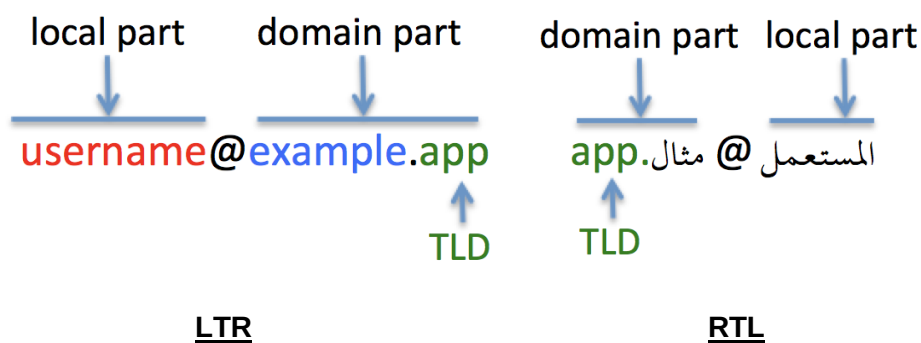
- 本地部分 (“@” 字符之前)。
- 域部分 (“@” 字符之后)。

因为无论从左到右 (LTR) 还是从右到左 (RTL) 的文字均可在电子邮件地址和域名中使用，应从文字方向性的角度理解“之前”和“之后”。

(虚构) EAI 地址示例

user@example.みんな	使用 IDN TLD
user@大坂.info	使用 IDN 二级域
用戶@example.lawyer	使用统一码 (Unicode) 本地部分和新 gTLD

EAI 地址中从右到左文字示例





在国际化电子邮件地址中，域部分可以包含任何域名（包括带有新 TLD 的域名），也可以包含统一码 (Unicode) U-标签。本地部分不属于域名，原则上可以包含几乎任何统一码 (Unicode) 字符，但实际上邮件系统会对邮箱名称中使用的字符做出限制。

国际化电子邮件地址 (EAI) 这个术语通常是指在电子邮件中使用国际化地址。

动态链接生成（链接化）

现代软件（如热门文字处理或电子表格应用程序）有时允许用户创建超链接，只需输入貌似网址、电子邮件地址或网络路径的字符串即可生成。例如，如果应用程序将 "www." 识别为特殊前缀或将 ".org" 识别为特殊后缀，则在电子邮件中输入 "www.icann.org" 可自动生成可点击链接 <http://www.icann.org>。

一经使用，链接化应采用统一方式处理所有（而不只是部分）格式正确的网址、电子邮件地址或网络路径。精准链接化很难实现，还需取决于文字背景（例如，在某些语言中，"www" 并非表示网址），此处不再过多赘述。

根区注册管理机构的动态属性

DNS 属于大型分布式数据库，分为多个部分，俗称区域。包含所有 TLD 的部分称为根区，因为从概念角度而言它位于 DNS 域名树的根部。所有 DNS 区域（包括根区）均会根据需要进行更新。增添新 TLD 或淘汰旧 TLD，相应地将新 TLD 名称添加到根区或从根区中删除旧 TLD。

这意味着，所有 TLD 固定列表（如存储于应用程序或文件中的列表）最终必然会过时。为切实验证域名中的 TLD，软件可以通过 DNS 查询执行在线检查；或者，如果使用文件，则可定期刷新文件。稍后将更详细地介绍这两项内容。

普遍适用性行动方案

普遍适用性的五大标准

普遍适用性是指所有支持互联网功能的应用程序、设备和系统以统一方式正确接受、验证、存储、处理以及显示全部有效域名和电子邮件地址的状态。普遍适用性的五大标准如下所述。

1. 接受²	接受是指以字符串的形式从用户界面、文件或者软件应用程序或在线服务使用的 API 中接收电子邮件地址或域名的过程。 应用程序和服务允许通过以下方式提供域名和电子邮件地址： ▪ 通过用户界面输入；或者 ▪ 通过 API 从其他应用程序和服务接收。
--------------------------	--

² 在本文件中，“接受”与“验证”有所不同。实际上，两种行为可能存在重叠。



2. 验证³	验证是指应用程序或在线服务以字符串的形式收发电子邮件地址或域名的过程中可能多次出现的一种行为。 验证旨在确保输入的信息有效，至少保证绝非无效信息。验证可确保信息采用正确的语法，还可进行其他检查。 很多程序员一向依靠特别验证方法（如检查 TLD 是否符合长度限制，或者字符是否来自 ASCII 字符集）对域名和电子邮件地址进行验证。然而，由于互联网环境不断变化，此类方法依托的假设条件不再适用： ▪ 域名和电子邮件地址现在可以包含非 ASCII 统一码 (Unicode) 字符。 ▪ TLD 列表不断变化。 ▪ 域名中的任何标签（包括 TLD 标签）的长度均不能超过 63 个字符。⁴ 还可以使用其他技术验证 TLD，稍后将进行介绍。
3. 存储	存储是指先以字符串的形式将电子邮件地址或域名存储到软件应用程序或在线服务使用的数据库或文件中并在此后使用相同或其他软件应用程序进行检索的过程。 应用程序和服务可能需要长期和/或短暂存储域名和电子邮件地址。无论数据生存期如何，必须采用以下格式存储： ▪ 互联网标准意见征询 (RFC) 定义的格式；或者（不够理想） ▪ 可与 RFC 定义的格式相互转换的另一种格式。 尽管 DNS 名称和电子邮件地址中的统一码 (Unicode) 存储为 UTF-8，但在传统代码中可能遇到其他格式。请参阅下文的“良好实践”部分。

³ 在本文件中，“接受”与“处理”有所不同。实际上，两种行为可能存在重叠。

⁴ 如果是 ASCII 标签，63 个字符的长度限制适用于标签本身；如果是 IDN，则长度限制适用于 A-标签格式标签。



4. 处理 ⁵	处理是指电子邮件地址或域名被应用程序或服务用于执行活动（如搜索或对列表排序）或者被转换为另一种格式（如由 U-标签转换为 A-标签）的过程。 在处理过程中，可能需要额外进行验证。电子邮件地址和域名处理方式完全取决于应用程序开发人员的想象力；但值得注意的是，切勿做出任何受 DNS 外部政策影响的假设（例如，发送给 <code>pākehā@tetaurawhiri.govt.nz</code> 的电子邮件收件人位于新西兰）。
5. 显示	显示行为是指在用户界面中呈现电子邮件地址或域名的过程。 如果底层操作系统支持名称或地址中所使用的文字，且字符串采用统一码 (Unicode) 存储，那么域名和电子邮件地址的显示过程一般（但并非总是）会非常简单。⁶如果不满足这些条件，则可能需要进行特定于应用程序的转换操作。此外，即使底层操作系统支持字符串，但若混合使用 RTL 和 LTR 文字或者文本整体方向性不明确，显示也可能很复杂。

用户场景

以上示例和定义可能令大家误以为普遍适用性仅与计算机系统和在线服务有关。但事实上，还与使用这些系统和服务的用户息息相关。

以下是一些需要普遍适用性的活动示例：

注册新 TLD	组织采用“品牌”TLD，通过提供以下格式的电子邮件地址为客户营造与众不同的客户体验：<code>customername@example.brand</code>。 普遍适用性是指： 网站和应用程序接受“<code>@example.brand</code>”一类电子邮件地址，如同使用旧 TLD（如 <code>.com</code>、<code>.net</code>、<code>.org</code>）一样。
---------	--

⁵ 在本文件中，“处理”与“验证”有所不同。实际上，两种行为可能存在重叠。

⁶ 务必认识到显示并不简单，即使某些复杂文字满足这些条件也不例外。



访问 gTLD	用户在浏览器中键入地址或单击文件中的链接，访问域名包含新 TLD 的网站。 普遍适用性是指： 即使是新 TLD，用户的浏览器也会以其本机格式显示网址，访问用户所期望的网站。除非在某方面对客户有利，否则浏览器不会以 A-标签格式向用户显示域名。
使用包含新 gTLD 的电子 邮件地址作为在线身份	用户获取域部分使用新 gTLD 的电子邮件地址，将该电子邮件地址用作访问其银行帐户和航空公司忠诚帐户的身份。 普遍适用性是指： 即使电子邮件地址中使用的域是新域，银行或航空公司网站也会像接受旧 TLD（如 .biz 或 .eu）地址一样接受新地址。
访问 IDN	用户在浏览器中键入 URL 或单击文件中的链接，访问 IDN URL。 普遍适用性是指： 即使域名包含的字符与用户计算机的语言设置不同，用户希望使用的各种浏览器也会按预期显示网址并成功访问网站。
在电子邮件中使用国际 化电子邮件地址	用户已获取新的电子邮件地址，该电子邮件地址的域名中包含非 ASCII 字符（例如，info@普遍接受-测试.世界）。 普遍适用性是指： 用户可以使用任何电子邮件客户端从任何电子邮件地址收发邮件。
使用国际化电子邮件地 址作为在线身份	用户获取非 ASCII 电子邮件地址，将该电子邮件地址用作访问其银行帐户和航空公司忠诚帐户的身份。 普遍适用性是指： 银行或航空公司网站完全接受新身份，就像对待其他任何电子邮件身份一样。
在应用程序中动态创建 超链接	用户在文件或电子邮件中输入网址。 普遍适用性是指： 如果地址为非 ASCII 地址或包含新 TLD，则应用程序采用相同的规则自动生成超链接。



开发应用程序

开发人员编写用于访问网络资源的应用程序。

普遍适用性是指：

- 开发人员使用的工具包含库，这些库通过支持新 TLD 和 IDN 实现普遍适用性。

违背国际惯例

以下做法认为不良实践：

✗	向用户显示 A-标签，但未展示相应的用户收益，如显示 U-标签与 A-标签之间的映射关系。
✗	注册新的电子邮件地址时，要求用户输入 A-标签；或者，注册新的托管域时，要求用户输入 A-标签。
✗	使用过时标准或非权威性在线域名资源验证域名或电子邮件地址的语法。
✗	即使定期添加及删除新 TLD，也会用到过时的 TLD 列表。
✗	向用户公开 A-标签的内部使用情况。 例如，在回复 EAI 用户时，将 EAI 地址中的域转换为 A-标签。
✗	将某些域名视为搜索词而不是域名，因为应用程序无法将其识别为域名。
✗	设置垃圾邮件阻止程序，在未论证是否存在滥用的情况下自动阻止全部（新）TLD。

UA 就绪技术要求

为使应用程序或网站达到 UA 就绪状态，必须满足各种要求。

整体要求

支持普遍适用性 (UA) 的应用程序或服务：

1. 支持所有域名，无需考虑长度或字符集。

请参阅 [RFC 5892](#)。

2. 允许所有对域名和电子邮件地址有效的字符集。

接受统一码 (Unicode) 码点及 ASCII。



3. 可以正确呈现统一码 (Unicode) 字符串中的所有码点。

请参阅 [RFC 3490](#)。请注意，统一码 (Unicode) 定期添加新码点，因而一直在变化。

4. 可以正确呈现从右到左 (RTL) 的字符串，如阿拉伯语和希伯来语字符串。

如需了解有关 RTL 文字的信息，请参阅 [RFC 5893](#)。

5. 可以使用 UTF-8（如有需要）及可与 UTF-8 相互转换的其他编码，在应用程序与服务之间开展数据通信。

如需了解有关 UTF-8 的信息，请参阅 [RFC 3629](#)。

6. 提供支持 UTF-8 的公共 API 和专用 API。

专用 API 仅适用于同一供应商的服务间调用。

7. 正确对待 EAI 地址。

特别是，切勿将地址中的 IDN 转换为 A-标签。

8. 可以向收件人发送电子邮件，也可以从收件人接收电子邮件，无需考虑域名或字符集。

请参阅 [RFC 6530](#)。

9. 使用支持统一码 (Unicode) 的格式存储用户数据，并且可以与 UTF-8 相互转换。

此类转换仅对产品或服务运营商可见。

10. 支持官方 ICANN TLD 列表中的所有顶级域名，无需考虑长度或字符集。

请参阅 <https://data.iana.org/TLD/> 上的官方列表。

开发人员注意事项

鉴于很多现有软件系统包含关于域和电子邮件地址的硬编码假设，可能需要更改代码才能识别 IDN、新 TLD 和 EAI 邮件地址。本部分讨论开发人员如何整合为实现普遍适用性而做出的代码更改。

设计符合兼容性和灵活性要求的软件

乔·波斯特尔 (Jon Postel) 在 [RFC 793](#) 中提出的鲁棒性原则，堪称软件开发的总体设计原则：

“对于己方的行为要保守，对于他方的行为要宽容。”

也就是说，保守对待您发送的内容：如遇任何规范含糊不清之处，避免任何可能令他人感到吃惊的信息。另一方面，在接收内容时，接受任何看似有效的信息。这并不是指通过更改代码解决其他实施举措中的明显错误，因为这会导致无据可循且“无法调试”的混乱局面。



通过开发和更新软件实现 UA 就绪的良好实践

接受

✓	尽量将名称显示为统一码 (Unicode)。 应允许（但不要求）用户以 A-标签（而不是 U-标签）格式输入域名。但默认情况下显示 U-标签，仅在具有一定收益时向用户显示 A-标签。
!	切勿生成包含 A-标签的 EAI 地址；但是，如果其他方的软件显示 A-标签，也可以进行处理。
✓	如果任何用户界面元素要求用户输入域名或电子邮件地址，必须接受长名称。ASCII 域名的每一个标签最多可以包含 63 个字符，总长度最多可达 253 个字节。UTF-8 标签可能比 ASCII 标签长得多，总长度最多可达 670 个字节。请记住，大部分统一码 (Unicode) 码点的 UTF-8 代码为 1 个字节以上。 ▪ 请参阅 RFC 1035。

验证

✓	仅在适当时执行验证。 仅在应用程序或服务操作要求时执行验证。这是确保全部有效域名均获得系统接受的最可靠方法。
✓	认识到语法正确的输入可能代表互联网目前正在使用的域名或电子邮件地址。可能有效，也可能无效，具体取决于应用程序。
!	验证时，请注意以下事项： ▪ 根据官方公布的表格，验证域名的 TLD 部分。IANA 发布了顶级域列表，网址为： * https://data.iana.org/TLD/tlds-alpha-by-domain.txt * 另请参阅：https://www.icann.org/en/system/files/files/sac-070-en.pdf ▪ 在 DNS 中查询域名。 * GETDNS API (http://getdnsapi.net/) 是极为便捷的 DNS 查询方法。 * 大部分操作系统还具有本机 DNS 查询 API。 ▪ 要求重复输入电子邮件地址以检测输入错误。



- 检查各标签是否遵循“国际化域名应用”(IDNA 2008) 规则, 对标签中的字符进行验证。
 - * 请参阅 [RFC 5892](#)
- 将对标签本身的验证限定于 RFC 中定义的少量完整标签规则。
 - * 请参阅 [RFC 5894](#)
- 确保产品或功能正确处理数字。
 - * 例如: 在数字输入字段中, 阿拉伯-印度文数字字符及 ASCII 数字应视为数字。
 - * 请注意, 阿拉伯-印度文数字在 U-标签中有效, 但在此类情况下不得视为等同于要 ASCII 数字。

存储

✓	应用程序和服务应支持最新统一码 (Unicode) 标准。
✓	应尽量使用 UTF-8 格式存储信息。 部分系统可能还要求支持传统 UTF-16 格式, 但通常首选 UTF-8。UTF-7 已弃用, 而 UTF-32 则过大, 不适用于文件存储。 酌情规范字符串 (在某些情况下, 规范化可能会导致信息丢失)。
!	存储时仔细考量所有端到端场景, 然后再转换 A-标签和 U-标签。 在新应用程序中, 最好仅在文件或数据库中使用 U-标签, 因为这样可以简化搜索、排序和表示。但与未启用统一码 (Unicode) 的旧应用程序和服务互用时, 转换可能会受到影响。
✓	标记存储中的电子邮件地址和域名以便于访问。 在文件的“作者”字段中填写电子邮件地址和域名或者在日志文件中填写“联系信息”会导致原始地址丢失。
✓	无论地址和域名的存储方式如何, 必须能够匹配多种格式的字符串。 例如, 搜索 example.みんな 应同时找到 example.xn--q9jyb4c。

处理

✓	确保所有网络服务器和 MIME 邮件回复的内容类型均指定了 UTF-8。
---	--------------------------------------



✓	在网络服务器 http 标头中指定 UTF-8。 ▪ 务必确保在每次回复时指定编码。
!	处理过程中，将 A-标签转换为 U-标签之前需考量背景，反之亦然。 最好仅在文件或数据库中使用 U-标签，因为这样可以简化搜索和排序。但与未启用统一码 (Unicode) 的旧应用程序和服务互用时，转换可能会受到影响。
✓	确保产品或功能根据区域设置和语言规范处理排序顺序、搜索和排序规则，同时确保支持多语言搜索和排序。
✗	切勿对域名中的标签使用百分号编码： ▪ example.みんな 正确 ▪ example.%E3%81%BF%E3%82%93%E3%81%AA 错误。
✓	鉴于统一码 (Unicode) 标准不断延伸，应检查创建应用程序或服务时未予定义的码点，避免生成错误或混乱输出。 底层操作系统字体缺失可能导致生成无法显示的字符（通常使用小方框表示此类字符），但此类情况不应触发故障或错误消息。
✓	使用受支持且启用统一码 (Unicode) 的 API。
✓	使用最新的 IDN 国际化域名应用 (IDNA 2008) 协议和表文件： ▪ RFC 5891 ▪ RFC 5892
✓	尽量使用 UTF-8 格式处理文本。
✓	协调升级应用程序及其所依赖的服务。 如果服务器是统一码 (Unicode) 服务器，客户端是非统一码 (Unicode) 客户端（反之亦然），则每次交易均需转换数据，这样不但容易出错，而且速度很慢。
✓	执行字符转换时，文本字符串长度可能出现大幅增减。每个 UTF-8 码点可能介于 1 到 4 个字节之间；在某些情况下，其他编码格式的单字符可能对应于若干 UTF-8 码点，反之亦然。



显示

✓	显示底层操作系统支持的所有统一码 (Unicode) 码点。 现代操作系统均支持统一码 (Unicode)，但其呈现引擎并不总能正确呈现所有文字和语言。仅当目标操作系统无法正确呈现时，才会在应用程序中提供字符呈现。
✓	开发应用程序或服务时，充分考虑支持的语言，确保操作系统和应用程序涵盖这些语言。
✓	显示前，将 A-标签转换为 U-标签。 例如，最终用户应看到 "example.みんな"，而不是 "example.xn--q9jyb4c"。（此转换是一个 UA 就绪处理示例）。
✓	默认情况下，将域名显示为 U-标签。 仅在具有一定收益时向用户显示 A-标签。
!	请注意，可以使用混合文字域名。 - 某些统一码 (Unicode) 字符肉眼看似相同，但计算机却可以发现不同；例如，拉丁文字母 O、西里尔文字母 О 和希腊文字母 Ο。 - 混合文字字符串在几种密切相关的文字（例如，日本汉字、片假名、平假名和日本罗马字）中很常见。否则，混合文字很可能被恶意利用，如网络钓鱼。应按照《统一码 (Unicode) 技术标准 39》中的“统一码 (Unicode) 安全机制”⁷来检查采用统一码 (Unicode) 序列的文字是否符合良好实践。 - 如果用户界面调用字符串以引起用户注意，请确保这种做法不会对非拉丁文字用户造成不利影响。 如需了解有关统一码 (Unicode) 安全注意事项的更多信息，请访问： http://unicode.org/reports/tr36。
✓	请注意未分配及禁止的域名字符。 请参阅 RFC 5892

⁷ 请参阅 https://www.unicode.org/reports/tr39/#Restriction_Level_Detection



统一码 (Unicode)

✓	使用受支持且启用统一码 (Unicode) 的 API。
✗	使用经过精心调试的标准 API： ▪ 字符串格式转换。 ▪ 确定哪种文字包含字符串。 ▪ 确定字符串是否混合使用文字。 ▪ 统一码 (Unicode) 规范化/分解。
✗	切勿使用 UTF-7，限制使用 UTF-32。 ▪ UTF-7 已弃用。 ▪ UTF-32 的每个码点使用四个字节。鉴于每个码点占用等量空间，而且可以直接在数组中创建索引，因此在程序代码中使用十分方便，但或许过于庞大，不适合存储到文件和数据库中。
✗	除非明确要求（如某些 Windows API 和 Javascript 应用程序），否则切勿使用 UTF-16。 在 UTF-16 中，16 位只能代表从 0x0 到 0xFFFF 之间的字符。超出此范围（0x10000 到 0x10FFFF）的值使用伪字符对，称为代理。如果未就代理对处理进行全面测试，则可能导致棘手错误乃至潜在安全漏洞。
✓	在 Cookie 中使用 UTF-8，以便应用程序正确读取。
✓	使用 IDNA 2008 协议和表文件： ▪ RFC 5891 ▪ RFC 5892
✗	切勿使用已被 IDNA 2008 取代的 IDNA 2003。
!	保持各个版本的 IDNA 和统一码 (Unicode) 表相互一致。 例如，除非应用程序实际执行表文件中输入码点解释的分类规则 (RFC 5892)，否则必须从系统支持的统一码 (Unicode) 版本生成 IDNA 表。这些表不需要显示最新版本的统一码 (Unicode)，但必须保持一致。
✓	使用 IDNA 2008 完整标签规则验证标签。 ▪ 在某些情况下，可能还要实施进一步验证；例如，应用程序明确了解其使用的域名中允许包含哪些文字。



一般规定

✓	使用权威性资源验证域名。 切勿做出过时的 <i>特别</i> 假设，如“所有 TLD 的长度均不超过 6 个字符。”
✓	确保产品或功能正确处理数字。 例如，在数字环境中，ASCII 数字和亚洲表意数字表示均应视为数字。
!	面向非预期位置查找疑似 EAI 地址的邮件地址： ▪ 艺术家/作者/摄影师/版权元数据。 ▪ 字体元数据。 ▪ DNS 联系记录。 ▪ 二进制版本信息。 ▪ 支持信息。 ▪ OEM 联系信息。 ▪ 注册、反馈及其他形式。
!	限制生成新域名和电子邮件地址时允许使用的码点： 所有使用电子邮件地址的产品必须接受国际化电子邮件地址，以便在本地部分使用大部分 UTF-8 打印字符。但是，当用户创建新的 IDN 或 EAI 地址时，应用程序或服务则无需允许使用所有这些字符。 首先，阻止创建特定 IDN 或电子邮件地址可减轻某些可能出现的安全性和可访问性隐患。（注意：如果提供字符串，为秉承良好实践原则，仍会要求软件接受此类字符串。）
!	请注意，绝不能总是单凭自动化测试用例来衡量普遍适用性。 例如，不一定总能测试应用程序或协议如何处理网络资源，有时最好通过功能规范和设计审核来验证是否符合要求。
!	切勿假设因某个组件不直接调用名称解析 API 或不直接使用电子邮件地址，因而不会受到影响。 了解组件如何获取域名 — 并非始终通过用户交互获取。以下是关于组件如何获取域名的一些示例： ▪ 组策略。 ▪ LDAP 查询。 ▪ 配置文件。 ▪ Windows 注册表。 ▪ 与其他组件或功能之间互相转移。



执行代码审核，避免缓冲区溢出攻击。

- 在统一码 (Unicode) 中，当大小写折叠或规范化时，字符串长度可能相应增减。
- 执行字符转换时，文本长度可能出现大幅增减。

其他挑战

检测和转换字符集的机制

一些旧电子邮件应用程序使用本地字符编码，无法根据需要进行检测 UTF-8 并与 UTF-8 相互转换。在处理电子邮件标头（收件人、抄送、密送、主题）时表现尤为明显。

将多个电子邮件地址作为一个用户身份管理

当用户有多个电子邮件地址时，分别将这些地址作为单个用户身份进行管理可能十分繁琐。

电子邮件程序可将发送至此类别名的流量定向至同一邮箱，但应用程序仍能将它们视为不同的身份。

域名权威性来源：DNS 根区和 IANA 列表

TLD 官方列表共有两个来源。第一个来源是 DNS 根区本身。经过域名系统安全扩展 (DNSSEC) 签名，因此尽管内容很难解析为文本文件，但可通过 DNSSEC 兼容名称服务器对内容进行验证。另一个来源是 IANA 发布的 TLD 文本文件（每行一个 TLD，按字母顺序排列）。此类文件位于 [https 网络服务器](https://www.internic.net/domain/root.zone)，因此下载时最好检查网站的安全传输层协议 (TLS) 证书是否有效，从而确保获取正确的文件。

您可从以下任一链接获取 TLD 列表：

- <https://www.internic.net/domain/root.zone>（根区文件）
- <https://data.iana.org/TLD/tlds-alpha-by-domain.txt>（文本 TLD 文件）

包含 IDN 的电子邮件以及为何它不同于 EAI

国际化电子邮件地址 (EAI) 邮件倾向于使用 UTF-8 域名；不建议使用 ASCII 编码 A-标签。一些邮件系统已针对包含 IDN 的电子邮件地址做出部分规定，但并未提供全方位 EAI 支持。由于 IDN 可以表示为 ASCII A-标签，因此某些现有软件允许使用 ASCII 或统一码 (Unicode) 表示电子邮件地址中的 IDN。例如，出于种种目的（发送、接收和搜索），某软件平等对待以下两个 IDN 地址：

`user@example.みんな` = `user@example.xn--q9jyb4c`

但是，即使两个地址均有效，某软件也不会将这些地址视为等效地址，因为在将 A-标签 ("xn--q9jyb4c") 转换为等效 U-标签 ("みんな") 之后才会进行比较。这可能导致用户体验不可预测。如果某软件出于“兼容性”需要将 U-标签转换为 A-标签，用户体验可能会变得尤其混乱。回复或转发邮件时用户看到的地址不同或者无法按预期搜索和排序地址的情况可能会随之增加。



如下所示，某软件可能尝试使用国际化域名编码 (Punycode) 转换电子邮件地址的本地部分，国际化域名编码 (Punycode) 算法用于将 A-标签转换为 U-标签（反之亦然）。此类转换无效，而且还会创建无法投递的无效地址。

切勿尝试将电子邮件地址的本地部分转换为其他形式

✓ 用戶@example.みんな

✗ xn--youq53b@example.xn--q9jyb4c

稳健的 UA 就绪软件和服务应能够正确处理和对待所有这些格式、处理地址中的 UTF-8 本地部分和 UTF-8 U-标签，同时还可以接受地址中的 A-标签以实现向后兼容性。

链接化及其挑战

现代软件有时允许用户自动创建超链接，只需输入貌似网址、电子邮件名称或网络路径的字符串即可生成。例如，如果应用程序将 "www." 识别为初始标签或将 ".org" 识别为 TLD，则在电子邮件中输入 "www.icann.org" 可自动生成可点击链接 <http://www.icann.org>。

链接化是指应用程序接受字符串并动态确定应创建指向互联网位置 (<http://> 或 <https://>) 还是电子邮件地址 (<mailto:>) 的超链接的操作。链接化（如果发生）应采用统一方式处理所有格式正确的网址、电子邮件名称和网络路径。

链接化使用软件开发人员创建的算法和规则，确定是否应将字符串解释为链接。与此相关的是：人们究竟如何将字符串识别为域名？尽管浏览器、电子邮件客户端和字处理器均具有此项功能，但还有很多其他应用程序可以做出相关判断。

良好实践建议

1. 尝试基于显式协议前缀（例如 "<https://>"、"<ftp://>"、"<mailto:>"）将文字转换为链接，但只有字符串的其余部分格式正确时才完成这项操作。

示例字符串	预期行为/结果
example.com	未创建链接，因为协议不存在且无法推断。
http://example.com	创建超链接，因为协议是显式协议。
http:example.com	未创建链接，因为语法错误 (missing //)。
http://example.a	未创建链接，因为 "a" 并非 TLD。
http://example..ab	未创建链接，因为语法错误（连续两个点）。
http://普遍接受-测试.世界	创建超链接，因为协议是显式协议。

2. 尝试基于隐式协议前缀（例如 "www" 推断 "<http://www>"）将文字转换为链接。



示例字符串	预期行为/结果
www.example.com	创建超链接，因为协议是隐式协议 ⁸
label@example.com	创建 <code>mailto: label@example.com</code> ，因为协议是隐式协议。

3. 围绕包含双向文本的 URL 的 HTML 可能包括影响文本显示方向的代码。链接化版本应保持同一显示方向。
4. 如果将 TLD 用作确定可链接性的“专用标记”，则必须包括所有 TLD。应频繁更新 TLD 列表。

统一码 (Unicode) — 背景和码点属性

自 1991 年首次发布 Unicode 1.0 以来，统一码 (Unicode) 标准持续发展。自此以后，每个版本都新增了更多的字符和码点，以便处理更多的语言和文字。当前版本是 12.1。

在统一码 (Unicode) 中，每个码点都包含一组属性，如 `Uppercase_Letter`、`Decimal_Number` 或 `Nonspacing_Mark`。许多字符（如拉丁文、汉语（中文）或阿拉伯文）具有文字属性，而其他字符（如标点符号）则没有文字属性。

如下所述，IDNA 使用码点属性确定 IDN 中允许使用哪些字符。统一码 (Unicode) 字符数据库 [UAX#44](#) 描述码点属性数据库。

UTF8、UTF16 及其他编码方法

统一码 (Unicode) 码点可包含介于 0 到 0x10FFFF 之间的数值。鉴于单字节只能包含介于 0 到 0xFF 之间的值，因而需要通过某种多字节编码存储统一码 (Unicode) 码点。

原始版本的统一码 (Unicode) 码点不足 64K (0xFFFF)，每个码点均可嵌入 16 位整数，继而形成双字节编码，称为 UCS 或 UCS-2。当统一码 (Unicode) 扩展超过 64K 码点后，UCS 扩展为 UTF-16⁹，使用 16 位码点对（否则无效，称为代理）表示大于 64K 的值。在这种情况下，鉴于代理会提高用于计算字符串码点数量或按码点顺序排序字符串的代码复杂度，从而引发调试问题。另外还面临一个问题：一些计算机（如 IBM 制造的计算机）先存储 16 位值的高字节（“大端”），另一些计算机（如 Intel 制造的计算机）则先存储低字节（“小端”）。因此，UTF-16 具有两个存储变体：UTF-16BE 和 UTF-16LE。有些技术可检测及修复字节序问

⁸ 注意：实际网站可能仅使用 `https`，需要 `https://` 而非 `http://`。当您遇到这种情况时，可能无法解析超链接。

⁹ 有关 UTF-8、UTF-16 和 UTF-32 的技术详细信息，请参阅统一码 (Unicode) 标准第 3.10 节，网址为：<https://www.unicode.org/versions/Unicode12.0.0/ch03.pdf>。



题，但可能会引发错误。目前，UTF-16 主要用于采用 Microsoft Windows API、Java 和 Javascript 语言的现有应用程序。

另一种编码方法是 UTF-8，UTF-8 将每个码点编码为 1 到 4 个字节的可变长度字符串。UTF-8 相较于 UTF-16 具有若干优点，其中包括统一码 (Unicode) 的 ASCII 子集编码为单字节，因此所有 ASCII 字符串均自动显示为 UTF-8 字符串。UTF-8 通常比等效 UTF-16 更紧凑，而且排序也更方便，由于 UTF-8 字符串按字节顺序排序，因而会自动按码点顺序显示。IDNA 和 EAI 均需采用 UTF-8 编码。

UTF-32 是一种简单格式，以 32 位整数存储各个码点。鉴于 UTF-32 数组中的码点可以直接创建索引，因而便于进行程序内部处理；但是，由于规模庞大，极少应用于存储领域。

IDNA - 简史和现状

国际化域名应用 (IDNA) 由 IETF 于 2003 年首次提出，目前称之为 IDNA2003¹⁰。其中包括两种算法：一种算法用于将统一码 (Unicode) 码点映射为域名标签的标准形式，称为 Nameprep；另一种算法用于将统一码 (Unicode) 码点标签编码为 ASCII，称为国际化域名编码 (Punycode)。Nameprep 包括转换功能，如将大写字母映射为小写字母。

在参与 IDNA 工作一段时间后，2010¹¹ 年 IETF 制定并发布了修订规范，称为 IDNA2008。IDNA2008 创建了 U-标签和 A-标签两个术语，同时删除了 Nameprep 步骤，建议应用程序执行与区域设置和应用程序环境相适应的映射。2011 年，RFC 6452 根据 Unicode 6.0 对 IDNA2008 进行了更新，IETF 仍将继续进行审核。

实际上，仍有太多实施工作依然沿用 IDNA2003。少数库使用为 IDNA2008 创建的表（如 IDNA2003 中包含的表）。除统一码 (Unicode) 标准中包含的标准大小写折叠和规范化规则以外，IDNA2008 不存在任何区域设置映射。

UTS46 统一码 (Unicode) IDNA 兼容性处理提供了少量映射，这是一个例外。指定是否应接受或映射已映射到 IDNA2003 但允许在 IDNA2008 中使用的一些常用字符。对于应用程序而言，根据 IDNA2008（而不是 IDNA2003）处理这些字符很重要；如果使用 UTS46，则应确保按兼容 IDNA2008 的方式进行使用。

测试用例

应广泛使用各种域名和地址对用于处理 IDN 和 EAI 邮件地址的软件进行测试。请参阅 [UASG 004 UA 就绪评估用例](#)了解一系列用例。

升级 EAI 软件

为符合 EAI 要求，必需升级邮件服务器、提交和交付软件、邮件用户代理和网页邮件，以及任何用于处理电子邮件地址和发送邮件的应用程序。

¹⁰ 请参阅 RFC [3490](#)、[3491](#) 和 [3492](#) 中的定义。

¹¹ 请参阅 RFC [5890](#)、[5891](#)、[5892](#)、[5893](#)、[5894](#) 和 [5895](#) 中的定义。



如需详细了解 EAI、面临的问题及其实施方法，请参阅 [UASG 012 国际化电子邮件地址 \(EAI\): 技术概述](#)。

高级主题

复杂文字

除需要自行创建字符串解析或显示库的开发人员以外，其他读者可能对复杂文字细节知识不那么感兴趣。尽管如此，本文件还是进行了概括说明，确保所有读者充分了解相关知识，一旦在用户体验中遇到此类情形，可以识别与复杂文字有关的代码错误。

对于网页和电子邮件中的格式化 HTML 文本，HTML 标准具有完善的复杂双向文本处理和显示功能，开发人员应了解并使用这些功能以呈现文本。请参阅 WHATWG HTML 标准的呈现部分¹²，以及 W3C HTML 标准的对应部分¹³。

从右到左语言和统一码 (Unicode) 一致性

当横向显示文本时，一些文字（如拉丁文和梵文）从左到右显示字符，另一些文字（如阿拉伯文或希伯来文）则从右到左显示字符。如果从右到左书写文字使用从左到右书写的数字，或者使用英语或其他使用从左到右书写文字的语言的嵌入字，文本也可以采用双向格式。

当横向文本方向不一致时，可能导致理解困难乃至歧义。为解决此类问题，有一种算法可以确定双向统一码 (Unicode) 文本的方向性。

目前制定了一组规则，应用程序须应用这组规则，以便生成正确的显示顺序，如统一码 (Unicode) 双向算法所示。我们通常将其称为“Bidi 算法”。

Bidi 算法

Bidi 算法描述软件应如何处理同时包含从左到右 (LTR) 和从右到左 (RTL) 两种顺序的字符的文本。分配给短语的基本方向¹⁴ 将决定文本显示顺序。基本方向可以从左到右，也可以是从右到左，同时还能定义字符序列的显示顺序。在本文件中，基本方向是从左到右，因此所有字符序列均以先左后右的顺序显示。

为确定某一序列是从左到右还是从右到左，统一码 (Unicode) 中的每个字符均具有相关方向属性。大部分字母的强类型（强字符）为 LTR（从左到右）或 RTL（从右到左），具体取决于所属的文字。强类型 RTL 字符序列将从右到左显示。这与环境基本方向无关。例如：

¹² 网址为：<https://html.spec.whatwg.org/multipage/rendering.html>

¹³ 网址为：<https://www.w3.org/TR/2018/WD-html53-20181018/rendering.html>

¹⁴ 在 HTML 中，基本方向要么沿袭文件的默认方向（即从左到右），要么由使用 "dir" 方向属性的最近父元素显式设置。



(LTR) 示例 - مثال (RTL)。

一行之中可以混合使用方向不同的文本。在这种情况下，**Bidi** 算法可根据每一个方向相同的连续字符序列单独确定方向。

在统一码 (Unicode) 中，空格及大部分标点符号并未设置强类型 LTR 或 RTL，因为两种类型的文字均可使用。因此，它们被归类为中性字符或弱字符。弱字符是指通常按一个方向使用的字符；但在某些情况下，也可以按另一个方向使用。此类字符的示例包括：

- 欧洲数字。
- 东阿拉伯-印度文数字。
- 算术和货币符号。
- 很多文字通用的标点符号，如冒号、逗号、句号和无中断空格。

在未设定背景的情况下，中性字符的方向性不确定。部分示例包括：

- 标签。
- 段分隔符。
- 大部分其他空格字符。

当中性字符介于方向类型相同的两个强类型字符之间时，也将假定该方向性。例如，两个 RTL 字符之间的中性字符本身也将被视为 RTL 字符，并具有扩展定向运行的效果：

- نطاق.مثال

即使两个强类型字符之间有多个中性字符，也将以同样的方式对待。

当空格或标点符号位于两个方向不同的强类型字符之间时，中性字符将视为与主导基本方向的方向相同。例如：

- مثال.example

请记住，本文件的基准序列是从左到右，所以 *example* 是二级域，مثال 是 TLD。

除非支持方向重写，否则在编码及输入数字时始终采用高位数字优先原则，同时按 LTR 呈现数字。弱方向性仅适用于确定整个数字的位置。

统一码 (Unicode) 技术报告 9 对 **Bidi** 算法进行了全面详细介绍。

Bidi 域名规则

一个 **Bidi** 域名是至少包含一个 RTL 标签。**Bidi** 域名规则（参阅 RFC 5893¹⁵）旨在限制域名码点，通过双向显示规则避免出现两个码点序列不同但显示相同的名称。

¹⁵ 国际化域名应用 (IDNA) 的从右到左书写文字，RFC 5893, <https://www.rfc-editor.org/info/rfc5893>



连接符

某些语言使用字母文字，通过两个字符（称为二合字母）编写单个音素。换言之，二合字母是两个一组表示单个发音（或音素）的连续字母。

英文二合字母示例

ch（例如，church）	th (then)	sh (shoe)
ph（例如，phony）	th (think)	gh (rough)

一些二合字母完全连接在一起，组成连字。在书写和排版中，连字是指两个或两个以上的字素或字母连接构成单个字形。例如与号 (&)，从两个相互连接的拉丁字母 *e* 和 *t* (“*et*”是指“与”) 演变而来。在英文排版中，*fi* 和 *ffi* 通常显示为连字。

如果连字和二合字母在使用给定文字的所有语言中具有相同的解释，统一码 (Unicode) 规范化通常可以分析差异并分别匹配。倘若具有不同的解释，则必须使用其他方法（可能需要在注册表级别选择）匹配，或者必须向用户说明不会发生匹配。RFC 5894¹⁶ 第 4.3 节通过示例对解释不同的情况进行了说明。统一码联盟 (Unicode Consortium) 提出了两大主要策略，旨在应用 Bidi 算法处理零宽连接符（称为 ZWJ 和 ZWNJ）后确定特定字符的连接行为。（要了解有关这些连接符的更多信息，请参阅 <http://www.unicode.org/L2/L2005/05307-zwj-zwnj.pdf>。）

- 生成连接符时，实施期间可重新参考原始后备存储，确定是否存在相邻的 ZWNJ 或 ZWJ 字符。
- 或者，实施期间也可以使用这些相邻字符关联的带外字符属性替换 ZWJ 和 ZWNJ，避免信息干扰 Bidi 算法，并在字符重新排列时保留信息。一旦应用 Bidi 算法，将可使用带外信息以确保正确生成连接符。

域名注册管理机构及任何其他允许创建域名的实体（例如，创建三级及更低级别标签的应用程序）必须遵循 Bidi 域名规则，以确保统一显示域名并防止域名混淆，继而避免被不法分子利用发动同形异义词攻击。

要了解有关连接符的更多信息，请参阅 RFC 5894 第 4.3 节。

同形异义词和易混淆字

同形异义词是指因大小和形状相似而看似相同或极易混淆的字符。混用拉丁文、西里尔文和希腊文时经常遇到此类情况。例如，拉丁字母 “o”（代码 U+006f）、西里尔文小写字母 “o”（代码 U+043e）和希腊文小写字母 “o”（代码 U+03bf）。在某些情况下，一种文字（如克罗地亚文小写字母 “lj”（代码 U+01c9）与两个字母 “lj”（代码 U+006c U+006a）也存在同形异义词。请参阅 <http://homoglyphs.net/> 上的表以获取更多示例。

¹⁶ 国际化域名应用 (IDNA)：背景、说明和理由，RFC 5894，
<https://www.rfc-editor.org/rfc/rfc5894.html#section-4.2>



为防止出现包含同形异义词的域名，注册管理机构应使用标签生成规则 (LGR)，将标签中的码点限定为一种文字或兼容文字的一组码点。每家注册管理机构均应针对其接受注册的每种文字制定 LGR¹⁷。

要了解有关统一码 (Unicode) 混淆检测安全机制的更多信息，请参阅：

- http://www.unicode.org/reports/tr39/#Confusable_Detection

要了解有关易混淆字和良好实践的更多信息，请参阅：

- M3AAWG 统一码 (Unicode) 滥用概述和教程
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf>
- M3AAWG 统一码 (Unicode) 滥用行为预防最佳实践
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf>

规范化、大小写折叠和字符串准备

统一码 (Unicode) 规范化有助于确定任意两个统一码 (Unicode) 字符串是否彼此等同，同时提供用于处理和存储字符串的标准形式。在统一码 (Unicode) 中，某些字符可通过若干代码序列表示。人们称之为统一码 (Unicode) 等价性。统一码 (Unicode) 提供两种等价性：

- 规范等价性
- 兼容性等价性

显示相同视觉特征的序列称为规范等价。此类序列在打印或显示时呈现相同的外观和含义。例如：

U+006E（拉丁文小写字母 "n"）后跟 U+0303（组合用
颚化符 "ö"） = ñ

U+00F1（西班牙文小写字母 "ñ"） = ñ

统一码 (Unicode) 将 NFC（规范化形式 C）定义为先规范分解再规范组合。这样可以在不改变文本外观的情况下使文本码点数目降至最低。值得注意的是，在此示例中，根据 IDNA2008 以上三个字符均有效。

兼容等值是一种外观不同但在某些情况下含义相同的序列。它是字符或字符序列之间的一种弱等价关系。例如：

¹⁷ IANA 在国际化域名 (IDN) 库的实践操作中发布了一系列注册管理机构 LGR，网址为：
<https://www.iana.org/domains/idn-tables>。



U+FB00 (连字 "ff") = ff
U+0066 U+0066 (两个拉丁文字母 "f") = ff

在上面的示例中，根据定义，码点 U+FB00 为兼容等值，但并非序列 U+0066 U+0066 的规范等值。规范等值序列也是兼容等值序列，但兼容等值序列不一定是规范等值序列。

值得注意的是，根据 IDNA2008，码点 U+FB00 无效。

统一码 (Unicode) 将 NFKC (规范化形式 KC) 定义为先兼容分解再规范组合。这样可将文本缩减为标准集码点，而且可能会改变其外观。例如，NFKC 将连字 "ff" 转换为两个字母 "f f"，将经线符号 `am`(U+33C2) 转换为四个字符 "a.m." (U+0061 U+002E U+006D U+002E)。

为避免因使用规范等值但字符序列不同而引起的互用性问题，W3C 建议对所有文本使用 NFC。

要查看可按任意规范化形式做出更改的所有字符的列表，请参阅：

<http://www.unicode.org/charts/normalization>。

其他注意事项：

- IDN 标签中的字符必须为 NFC 格式。
- 当两款应用程序共享统一码 (Unicode) 数据但规范化方式不同时，可能会发生错误和数据丢失。
- 统一码联盟 (Unicode Consortium) 主张规范化形式必须长期保持稳定。换言之，为实现向后兼容，所有未来版本统一码 (Unicode) 的字符串必须保持规范化。
- 如前所述，在确定域名中允许使用的码点时务必持保守态度。

软件开发人员提示

- | | |
|---|--|
| ✘ | 切勿尝试通过转换为大写字母或忽略非空格字符来实现规范化，因为这会导致排序、数据复制导入和导出及客户端应用程序数据检索变得异常困难，而且还可能引发数据丢失或损坏。 |
| ✘ | 切勿在域名中使用 IDNA2008 禁止的码点。 |

要了解有关统一码 (Unicode) 规范化的更多信息，请参阅：

- <http://www.w3.org/TR/charmod-norm>
- <http://unicode.org/reports/tr15>

大小写折叠和映射

大小写折叠和映射是指将字符串中的所有字符转换为同一大小写形式（通常为小写）的过程。将大写字母 [A-Z] 映射成小写字母 [a-z] 适用于仅 ASCII 文本文件，但在使用其他字符的语言



中，情况恐怕要复杂得多。大小写映射可能与背景息息相关，映射的字符取决于映射背景，例如希腊文 **sigma** 的各种形式。也可能与区域设置关系密切，映射的字符取决于解释文本的区域设置，例如土耳其语中带点和不带点的大小写字母 **I**。大小写折叠与区域设置无关，尽管通过软件解释的字符串的大小写映射与区域设置存在一定关联，但其最终目的是生成供人类阅读的文本。最后，映射成大写字母与映射成小写字母**非**反函数。

对于 IDN，IDNA2008 允许应用程序使用任何适宜的大小写映射，因为映射发生在码点验证之前。实际上，不存在特定于区域设置的标识符映射，全体用户均使用统一码 (Unicode) UTS 46 映射¹⁸。

软件开发人员提示

✓	尝试执行大小写映射之前，请思考预期目标：这是标签、已知语言字符串还是其他类型的类属映射？
✓	在大小写折叠之前，先执行统一码 (Unicode) 规范化。

术语表和其他资源

术语表

A-标签	DNS 协议内部使用的国际化域名标签的 ASCII 兼容编码 (ACE) 表示形式。A-标签始终以 ACE 前缀 "xn--" 开头。A-标签可以转为 U-标签，U-标签也可以转回 A-标签，而不会丢失信息。
ACE 前缀	ASCII 兼容编码前缀 "xn--"。
ASCII	美国信息交换标准码。ASCII 包括无重音拉丁字符和欧洲阿拉伯数字。ASCII 属于统一码 (Unicode) 的一部分：各 ASCII 字符同样也是统一码 (Unicode) 字符。
API	应用程序编程接口 (API) 是一组用于构建软件 and 应用程序的例程、协议和工具。API 适用于基于网络的系统、操作系统或数据库系统，旨在促进使用给定编程语言为相应系统开发应用程序。
代码空间	定义编码上限和下限的范围。

¹⁸ UTS46 统一码 (Unicode) IDNA 兼容性处理，<https://www.unicode.org/reports/tr46/#Mapping>



码点	码点是代码空间中的数字值。码点用于区分数字值与其位序列编码，以及区分抽象字符与其特定图形表示形式（字形）。
DNS 根区	根区是 DNS 中央目录，也是在 DNS 中查找信息的关键组件。例如，将主机名转换为 IP 地址。
EAI	国际化电子邮件地址允许在电子邮件地址中使用 UTF-8 字符 — 域名、本地部分或二者兼有。
IANA	互联网号码分配机构。职能包括： ▪ 管理 DNS 根、.int 和 .arpa 域名以及 IDN 实施资源。 ▪ 协调全球 IP 和 AS 号码池，主要将这些资源提供给地区互联网注册管理机构 (RIR)。 ▪ 联合标准机构共同管理互联网协议的编号系统。
ICANN	ICANN 的使命在于确保全球互联网的稳定、安全与统一。要在互联网上访问另一个人的信息，您必须在电脑或其他设备中键入一个地址——可以是一个名称或是一串数字。这个地址必须是独一无二的，只有这样电脑之间才能互相识别。ICANN 负责协调这些分布在世界各地的唯一标识符并提供相应支持。ICANN 是一个非营利性公益机构，成立于 1998 年，其社群的参与者遍布世界各地。
IDN	国际化域名。IDN 指的是包含除 26 个基本拉丁字母 "a-z"、数字 0 到 9 和连字符 "-" 之外的 UTF-8 字符的域名。
IDNA	国际化域名应用。
IDN ccTLD	包含除 26 个基本拉丁字母 "a-z" 之外的字符的国家和地区顶级域。 示例： ▪ .рф（俄罗斯） ▪ .صر（埃及） ▪ .السعودية（沙特阿拉伯）
IETF	互联网工程任务组 (IETF) 是一个开放的大型国际社群，由与互联网架构的发展和互联网的平稳运行相关的网络设计人员、运营人员、供应商和研究人员组成。任何感兴趣的个人都可以加入此社群。IETF 负责制定互联网标准，特别是与互联网协议 (TCP/IP) 及网络协议（例如，HTTP 和 TLS）相关的标准。
语言	人类的口头或书面交流方法，包括以结构化方式和常规方式组合使用单词。



国际化域名编码 (Punycode)	通过域名系统 (DNS) 支持的有限 ASCII 字符子集表示 UTF-8 的算法。国际化域名编码 (Punycode) 在国际化域名应用 (IDNA) 框架的 A-标签中使用。
注册服务机构	供用户注册域名的组织。注册服务机构负责记录联系信息并将技术信息提交至中央目录机构（称为“注册管理机构”）。
注册管理机构	官方主数据库，其中包含在每个顶级域 (TLD) 内注册的所有域名。
RFC	意见征询 (RFC) 是互联网工程任务组发布的 (IETF) 正式文件，它是经委员会起草并经各方后续审核得出的结果。部分（但并非全部）RFC 记录批准的互联网标准。
文字	书面使用的字母或字符集合，代表一种语言的发音。
二级域名	在域名系统 (DNS) 层次结构中，二级域 (SLD 或 2LD) 是指仅次于顶级域 (TLD) 的下一级域。例如，在 example.com 中，example 是 .com TLD 的二级域。
U-标签	U-标签是由统一码 (Unicode) 字符组成且符合 IDNA 要求的字符串，至少包含一个非 ASCII 字符。U-标签可以转为 A-标签，A-标签也可以转回 U-标签，而不会丢失信息。
UA 就绪软件或 UA 就绪	支持同等接受、存储、处理、验证和显示所有顶级域、IDN 和电子邮件地址的软件。
统一码 (Unicode)	一种通用字符编码标准。其中定义了各字符在文本文件、网页及其他类型的文件中的表示方式。统一码 (Unicode) 旨在支持全球所有语言的字符。它可以支持大约 1,000,000 个字符。 请参阅： http://unicode.org 。
UTF	统一码 (Unicode) 转换格式。这是将统一码 (Unicode) 码点表示为字节流的一种方式。UTF-8 是处理 IDN 和 EAI 的首选 UTF。UTF-8 可将统一码 (Unicode) 转换为 8 位字节。
M3AAWG	信息传递、恶意软件和移动反滥用工作组 (M ³ AAWG) 是业内人士汇聚一堂协商对抗僵尸网络、恶意软件、垃圾邮件、病毒、DoS 攻击及其他在线攻击的场所。请参阅： https://www.m3aawg.org/ 。
W3C	万维网联盟 (W3C) 是一个国际社群，广大 <u>成员组织</u> 、 <u>全职工作人员</u> 和公众汇聚于此共同制定 <u>网络标准</u> ，如 HTML。请参阅： https://www.w3.org/ 。



WHATWG	网页超文本应用技术工作组 (WHATWG) 是由大批热衷通过标准和测试推动网络进步的热心人士组建的社群。 W3C 研讨会结束后，2004 年 Apple、Mozilla 基金会和 Opera Software 人员发起成立了 WHATWG。请参阅 https://whatwg.org/。
ZWJ	零宽连接符是一种用于对某些文字（包括阿拉伯文及所有印度文字）进行计算机排版的非打印字符。如果置于两个字符之间（否则无法连接），ZWJ 将可按连接形式打印这两个字符。
ZWNJ	零宽非连接符是一种用于对使用连字的书写系统进行计算机化的非打印字符。对于某些语言和文字而言，书写单词时许多字母自然会与下一个字母相连，从而形成连字。但是，为正确显示某些前缀、后缀和复合词，使用 ZWNJ 重写这种连接字母的自然行为，防止它们连接后面的字母（而不必在两个字母之间添加空格）。

如需完整 ICANN 术语表，请访问：<https://www.icann.org/icann-acronyms-and-terms/>。

RFC 和关键标准

IDN RFC	
RFC 3492	国际化域名编码 (Punycode): 一种适用于国际化域名应用 (IDNA) 的统一码 (Unicode) Bootstring 编码 RFC 3492 将国际化域名编码 (Punycode) 描述为： “一种简单高效的传输编码语法，专门用于国际化域名应用 (IDNA)” 国际化域名编码 (Punycode) 可以将统一码 (Unicode) 字符串转变为 ASCII 字符串，这种转变具有唯一性和可逆性。本 RFC 规定了一个称作 Bootstring 的通用算法，该算法允许使用一个由基本码点组成的字符串来唯一地表示从较大字符集中抽取的任意码点字符串。 https://tools.ietf.org/html/rfc3492



RFC 5890	国际化域名应用 (IDNA): 定义和文件框架 本 RFC 介绍了国际化域名应用 (IDNA) 修订版适用的使用条件及其所应遵循的协议。 https://tools.ietf.org/html/rfc5890
RFC 5891	国际化域名应用 (IDNA) 协议 本 RFC 规定了国际化域名应用 (IDNA) 协议机制, 依照该机制, 无需更改域名系统 (DNS) 本身, 即可注册并查询 IDN。 https://tools.ietf.org/html/rfc5891
RFC 5892	统一码 (Unicode) 码点和国际化域名应用 (IDNA) RFC 5892 指定了一系列规则, 用于确定是否应将某个码点 (无论是独立存在还是位于上下文中) 包含在国际化域名 (IDN) 中。 https://tools.ietf.org/html/rfc5892
RFC 5893	国际化域名应用 (IDNA) 的从右到左书写文字 本 RFC 提供了一个新的适用于国际化域名应用 (IDNA) 标签的 Bidi 规则, 旨在规范国际化域名中从右到左书写文字的使用。 https://tools.ietf.org/html/rfc5893
RFC 5894	国际化域名应用 (IDNA): 背景、说明和理由 本信息文件概要介绍了支持更高版本统一码 (Unicode) 的修订系统, 并提供了有关其系统组件的说明材料。 https://tools.ietf.org/html/rfc5894
RFC 5895	国际化域名应用 (IDNA) 2008 版中的字符映射 本 RFC 将介绍在实施中从收到用户输入内容到将允许的码点传递到新 IDNA 协议 (2008) 期间可采取的操作。其中描述了一种适用于用户输入的操作方法, 以便将相应用户输入应用至“网络”协议。另外还包括一套通用映射实施程序。 https://tools.ietf.org/html/rfc5895



EAI RFC	
RFC 6530	国际化电子邮件概述与框架 本标准将介绍一系列规范，以用于定义为充分支持国际化电子邮件地址而需要的各种机制和协议扩展。本文件将介绍如何组合运用国际化电子邮件地址的各个元素，以及邮件传输、标头格式和处理主要相关规范之间的关系。 https://tools.ietf.org/html/rfc6530
RFC 6531	适用于国际化电子邮件的 SMTP 扩展 本文件定义了简单邮件传输协议扩展，以便服务器展现接受及处理国际化电子邮件地址和国际化电子邮件标头的能力。 https://tools.ietf.org/html/rfc6531
RFC 6532	国际化电子邮件标头 本文件指定了对互联网邮件格式和 MIME 所做的改进，以允许在电子邮件地址和大多数标头字段内容中使用统一码 (Unicode)。本文件指定了对互联网邮件格式 (RFC 5322) 和 MIME 所做的改进，允许直接使用 UTF-8，而不仅限于在标头字段值（包括邮件地址）中使用 ASCII。针对使用此扩展格式的邮件定义了新介质类型 message/global。本规范还取消了禁止在邮件顶级类型子类型上使用非身份标识内容传输编码的 MIME 限制，以便通过现有邮件基础设施安全传输 message/global 部分。 https://tools.ietf.org/html/rfc6532
RFC 6533	国际化电子邮件传递状态与处理通知 本规范新增了一个适用于国际电子邮件地址的地址类型，以便系统能够正确保留含有非 ASCII 字符的原始收件人地址，即使在降级处理后也依然如此。此外，本规范还更新了传递状态通知和邮件处理通知的内容返回媒体类型，以支持使用新增的地址类型。 https://tools.ietf.org/html/rfc6533
RFC 8398	X.509 证书中的国际化电子邮件地址 本文件定义了一种新的名称形式，旨在纳入 X.509 使用者可选名称和颁发者备用名称扩展名的 otherName 字段，以便关联证书使用者与国际化电子邮件地址。 https://tools.ietf.org/html/rfc8398。



RFC 8399

RFC 5290 国际化更新

本档中介绍的 [RFC 5280](#) 更新与 2008 版国际化域名 (IDN) 规范保持一致，还在 X.509 证书中新增了国际化电子邮件地址支持。

<https://tools.ietf.org/html/rfc8399>

关键标准

ISO 10646 (统一码 (Unicode))

为制定通用技术规范以有效处理各种语言的电子信息，国际标准化组织 (ISO) 开发了一套国际编码标准，称为 ISO 10646。ISO 10646 颁布了一项适用于全球所有主要语言（包括繁体汉字和简体汉字）的统一字符编码标准。人们将这个大数据集命名为通用字符集 (UCS)。统一码 (Unicode) 标准制定了相同的字符集，而且进一步定义其他字符属性及实施者渴望了解的其他应用细节。

统一码 (Unicode) 是由统一码联盟 (Unicode Consortium) 设计的字符编码系统，旨在支持交换、处理和显示全球所有主要语言的书面文本。ISO 10646 和统一码 (Unicode) 针对通用字符库表定义了若干编码形式：UTF-8、UCS-2、UTF-16、UCS-4 和 UTF-32。

http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63182

GB18030 (中国)

GB 18030-2000 是一项中国政府标准，其中指定了继 UTF-8 之后可供中国市场使用的一种扩展代码页。字库表内部处理代码可以（而且应当）是统一码 (Unicode)；但该标准规定，软件提供商必须保证 GB18030 与内部处理代码能够成功相互转换。现阶段中国出售或将要出售的所有产品必须制定代码页迁移计划，从而支持 GB18030（无一例外）。GB18030 属于“强制性标准”，中国政府对认证过程做出了规定，以期巩固 GB18030 部署。

<http://icu-project.org/docs/papers/unicode-gb18030-faq.html>



在线资源

API	Windows 应用程序编程接口 (API) https://www.msdn.microsoft.com/enus/library/windows/desktop/ff818516%28v=vs.85%29.aspx SharePoint API https://msdn.microsoft.com/en-us/library/office/jj860569.aspx 公共域名后缀列表 https://publicsuffix.org/list/public_suffix_list.dat ICANN TLD 官方列表 http://data.iana.org/TLD/tlds-alpha-by-domain.txt Android API http://developer.android.com/guide/index.html MAC IOS API https://developer.apple.com/library/mac/navigation .Net Framework https://msdn.microsoft.com/en-us/library/system.text.encoding(v=vs.110).aspx
统一码 (Unicode) 安全	统一域名编码 (Unicode) 安全考量 http://www.unicode.org/reports/tr36 统一码 (Unicode) 安全机制 http://www.unicode.org/reports/tr39
统一码 (Unicode) 字符分组	统一码 (Unicode) 码平面 https://www.unicode.org/versions/Unicode12.0.0/ch02.pdf; 第 44-54 页 GB18030 概述 http://icu-project.org/docs/papers/gb18030.html BG18030-2000 与统一码 (Unicode) 官方映射表 http://source.icu-project.org/repos/icu/data/trunk/charset/data/xml/gb-18030-2000.xml 统一码 (Unicode) 规范化 https://unicode.org/reports/tr15/



统一码 (Unicode) 漏洞利用

《统一码 (Unicode) 技术报告 36》第 3.1 节 “UTF-8 漏洞利用”
http://unicode.org/reports/tr36/#UTF-8_Exploit

M3AAWG 统一码 (Unicode) 滥用行为预防最佳实践
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf>

M3AAWG 统一码 (Unicode) 滥用概述和教程
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf>

另请参阅：
<http://www.unicode.org>

其他

URI
<http://tools.ietf.org/html/rfc3986>

域名系统：非技术性说明 — 为什么通用解析性至关重要
<http://www.internic.net/faqs/authoritative-dns.html>

ICANN 术语表
<https://www.icann.org/icann-acronyms-and-terms/>

需要更多信息？

普遍适用性指导小组 (UASG) 和社群随时准备为软件开发人员和实施人员提供建议。

- ☐ 联系我们，就本主题发表看法及提出建议：info@uasg.tech。
- ☐ 加入普遍适用性讨论清单：<http://tinyurl.com/ua-discuss>。
- ☐ 要了解有关这项工作的更多信息，请访问：<http://www.icann.org/universalacceptance>。