Programming Language Hacks

UA103
Version 17-10-12
By Don Hollander
Secretary General
Universal Acceptance Steering Group

Universal Acceptance is the simple concept that all legitimate domain names and email addresses work across all applications.

A recent Universal Acceptance Steering Group (UASG) (http://www.uasg.tech) study found that many users were being denied access to applications because they lacked a simple fix.

Top-level domains (TLDs) and email addresses have evolved markedly since 2010, when non-ASCII characters were first introduced. Hundreds of these new style TLD names, including TLDs longer than three characters, have been added into the root zone. In 2012 non-ASCII characters became available in mailbox portion of email addresses.

Style of Address	Example Test Case
ascii@ascii.newshort	info1@ua-test.link
ascii@ascii.newlong	info2@ua-test.technology
ascii@idn.ascii	info3@普遍接受-测试.top
ascii@ascii.idn	info4@ua-test.世界
Unicode@ascii.ascii	测试1@ua-test.link
Unicode@idn.idn	测试5@普遍接受-测试.世界
Arabic.arabic@arabic	دون@رسيل.السعودية

In a recent study of 1000 popular websites, too few accepted the full range of email addresses to be used as unique identifiers. We found no consistency in the programming of the Regular Expressions used to validate email addresses and very little use of competent server-side libraries for validation, contributing to these poor results.

The UASG was established in 2015 to raise awareness of issues like this and to facilitate resolution. It is an initiative of the Internet community and is supported by ICANN. The UASG has developed a range of documentation and resources for becoming UA-ready, for both management and developers.

Developers must update their code to accommodate this growing number of domain names and email addresses. Here is some guidance for modernizing your applications:

Input
Data fields that accept domain names or email addresses must accept ASCII and non-ASCII characters. Many of the next billion Internet users to come online (and existing users that prefer addresses that better reflect their sense of identity) require text that doesn't use only ASCII. UTF-8 is the key here. This will affect input, storage and output of data from keyboards, databases and other data sources. Most modern software components are capable of supporting this. They just need to be configured correctly.

Validation
The easiest way to deal with this is to use a simple syntactic[1] validation of the email address in the client side and more extensive validation through server-side libraries. There are other ways of making sure the data entered is what the user meant, such as requiring entry of the field twice and doing a compare or sending an email to verify receipt. Using extensive and complicated Regular Expressions are often difficult to debug and may not cater to the now dynamic set of top-level domain names.

If you need to validate further, use a DNS lookup – that's the most certain. Or if you're going to use a local table of TLDs, make sure that it's from an authoritative source[2] and that your local table is updated at least daily.

Storage
The easiest way to deal with storage is to support Unicode. This ensures that the data is reproducible exactly as received. But for applications or systems that can't, there is an algorithm (Punycode)[3] that allows transformation of domain names between ASCII and non-ASCII strings.

Processing
When processing or sorting, it's important that equivalent names are treated as equivalent. Examples of equivalent but different representations include Unicode vs. Punycode, Unicode Normalization and the use of different native scripts. Treating equivalences will require some policies for the application or indeed the organization.

Display
Public-facing applications should be capable of displaying TLDs and email addresses in native scripts with appropriate fonts.

Validation Libraries
Programming language libraries, particularly open source programming language libraries, are creating or correcting validation routines, so becoming UA-ready may be as simple as re-compiling the code using the latest version of the library. The UASG is encouraging remediation work in many libraries.

When systems are UA-ready, they will work with the continuously expanding domain name space. It also sets businesses up for future opportunities and success by supporting their customers using their customers' chosen identities. It's time to get applications up to scratch.

For more information on this topic or about the UASG, or to contact us, please visit www.uasg.tech

---

About the author: Don Hollander is a New Zealand based former CIO for very large domestic and international corporations. He has been involved in the New Zealand IT industry for many years and served as the Chair of TUANZ in the 1990s and Chair of the 2020 Trust during the first decade of this century.

[1]Make sure there's one and only one '@', no consecutive dots '.', and the entire TLD length is less than 253 characters.

[2]There are a few options for the authoritative list of TLDs. The first option is the DNS root zone itself. It is DNSSEC-signed, so the list is properly authenticated. You can obtain the root zone from any of the following links:

- http://www.internic.net/domain/root.zone
- http://www.dns.icann.org/services/authoritative-dns/index.html
- http://data.iana.org/TLD/tlds-alpha-by-domain.txt

[3]https://www.ietf.org/rfc/rfc3492.txt

編程語言攻略

UA103
Version 17-10-12
Don Hollander，UASG工作組總秘書長

Universal Acceptance （簡稱UA，內地稱為「普遍接受」），是個關於讓所有合法域名及電郵地址應用於任何應用程式的的簡單概念。根據UASG工作組（Universal Acceptance Steering Group）最新研究顯示，很多用戶的域名及電郵因欠缺一項簡單修正而被拒登入應用程式系統。

2010年起，當非 ASCII 字元首次被採納作為頂級域名(TLDs) 及電郵地址登記服務後，業界發展一日千里，數以百計的創新頂級域名相繼誕生，如多於三個字符串的制式(.club, .website, .fashion等) 均進入頂級域名系統的根域檔案。 2012年，非ASCII字元亦成為電郵地址中使用者名稱的選項之一(註)。
註：電郵位址的格式為使用者名稱@主機名（域名）

根據UASG工作組提供的測試用例包括：

電郵地址類型	測試範例
ascii@ascii.newshort	info1@ua-test.link
ascii@ascii.newlong	info2@ua-test.technology
ascii@idn.ascii	info3@普遍接受-测试.top
ascii@ascii.idn	info4@ua-test.世界
Unicode@ascii.ascii	测试1@ua-test.link
Unicode@idn.idn	测试5@普遍接受-测试.世界
Arabic.arabic@arabic	دون@رسيل.السعودية

據最近一個名為「1000個最受歡迎網站」的研究顯示，只有少數網站接受所有類型的電郵地址作為唯一識別標示(unique identifiers)。我們發現不同網站在驗証電郵地址所使用的正則表達式(Regular Expressions)並不一致，並且甚少使用合乎水準的伺服器端函式庫(server-side libraries)作驗証，導致驗証效果不理想。

成立於2015年，UASG 工作組旨在加強有關互聯網系統普遍接受性及相關議題的認識，並促進業界尋求解決方案。UASG工作組由互聯網業界社群發動，全球域名協調管理機構ICANN 全力支持，公眾可透過官網查閱相關指引及資源(https://uasg.tech/documents)，同時歡迎業界管理人員及程式開發商一同加入「普遍接受性」(UA-Ready) 行列。

事實上，程式開發商必需定期留意及更新其應用程式編碼，以配合創新域名及電郵地址業務持續發展。關於現代化應用程式的指引包括如下：

輸入

接受域名或電郵地址的輸入欄位必需能接受/納ASCII及非 ASCII 字元。未來數以十億的互聯網用戶（及現時偏向以電郵地址標示個人身份的用戶）上網時再不需被限定使用ASCII字元而設的域名或電郵。此外，UTF-8字元輸入將成方便之匙。由於大部分現代軟件組件均支援UTF-8字元，只需配合適當的設置，便可在鍵盤、 數據庫及各數據來源，以UTF-8字元輸入、儲存及輸出數據。

驗証方式

在客戶端就電郵地址使用簡單句法驗証（註）[1]，同時在伺服器端透過函式庫來加強驗証是現行最簡便的驗証方式。此外，用戶亦可以其他方式確保輸入的數據合符用戶所指，如要求用戶端提供二次驗証作對比，或透過發送確認電郵作為驗証收據。若只通過使用廣泛而複雜的正則表達式，往往難以偵測及糾正程式中的錯誤，同時未必能符合現時頂級域名的動態組合。

如你想進一步驗証，可利用最常見的查找域名系統。或當你準備使用本地的頂級域名列表時，請確保一、該本地頂級域名列表是具權威性的查找來源（註）[2]及二、每日更新本地列表。

儲存數據

支援Unicode可算是現時處理儲存數據的最簡易方法。這確保數據被百分百接收作複製備份。然而，對於某些未能支援Unicode的應用程式或系統，它們可選擇域名代碼（註）[3]以確保域名在ASCII與非ASCII字元群組轉換的一致性。

處理

在處理及排列過程中，確保字符編碼的等價性可謂非常重要。例如視覺上不同字符形式展示但沒有語義上有區別的請況；如Unicode字符編碼對應域名代碼、Unicode正規化及不同原生編程語言等。維持字符編碼的等價性需符合某些應用程式或程式供應商的政策。

顯示器

對於面向公眾的應用程式，不同的原生編程語言應使用合適字型，以正確顯示頂級域名及電郵地址。

驗証函式庫

很多程式設計語言函式庫，特別是開源程式設計語言函式，附有開發及更正例程，所以兼容UA準則可簡單透過重新編譯最新版本以達到目的。同時UASG工作組正鼓勵大部分函式庫作者進行修正工程。

當系統能兼容UA，自能有助域名版圖持續擴展。而透過支持客戶選擇註冊使用突顯其個人身份的網域名，將間接為業界未來奠下更多商機/契機。就讓我們攜手協助這些應用程式標準更上一層樓！

如欲了解更多關於UASG工作組或其內容，歡迎瀏覽www.uasg.tech或與我們聯絡。

關於撰稿人：以新西蘭為基地的 Don Hollander ，過去曾在當地及不少國際機構擔任首席信息官(CIO)等要職。他在當地從事資訊科技業多年，於 90年代擔任TUANZ的主席，期後在 2000-2012年間擔任 2020Trust的主席。

註1： 請確保電郵地址只有唯一一個'@'、沒有連續'.'及整個頂級域名總長不多於253個字元。

註2：頂級域名有數個具權威性的查找來源表。第一選擇為域名解析系統的根域名伺服器。由於它已完成域名系統安全擴充簽章，故資料已是通過完整驗證。你可透過以下其他連結取得根域檔案:

- http://www.internic.net/domain/root.zone
- http://www.dns.icann.org/services/authoritative-dns/index.html
- http://data.iana.org/TLD/tlds-alpha-by-domain.txt

註3：https://www.ietf.org/rfc/rfc3492.txt