

Email addresses and domain names are *non*-latin! Now what?

Jim DeLaHunt / IUC41 / 16 October 2017

Internationalized domain names, email addresses

+1,000,000,000

The next one billion internet users use a wide variety of languages and scripts. Standards allow email addresses, and domain names, in scripts they can easily read. This is an introduction to those standards.

http:// 普遍接受 - 测试。世界

To: données@fußballplatz.technology

To: مانیش @ اَشوگا. لهند

Agenda

Slides: <http://go.jdlh.com/iuc41t4t1>

- * Who we are: UASG, Jim DeLaHunt
- * Context: the next one billion, and universal acceptance
- * So many top-level domain names!
 - * Exercises
- * Internationalized Domain Name for Applications (IDNA)
- * Email Address Internationalization (EAI)
- * Next steps
- * Q&A

Who we are



Who we are

Universal Acceptance Steering Group (UASG)

- * <http://www.uasg.tech>
- * Community-led initiative, world-wide
- * Raise awareness, identify problems, solve them
- * Project of ICANN, the domain name system organisation

Jim DeLaHunt

- * <http://jdlh.com>, ☎ +1-604-376-8953
- * Vancouver, Canada
- * Consultant in multilingual websites; software engineer
- * UASG volunteer participant

UASG materials available

UASG operates primarily by public education. Participants write outreach materials, technical notes. They give presentations to industry meetings. They evaluate, report, and follow up on UA issue reports.

Technical Notes (selection)

- * UASG004 Use Cases for UA Readiness Evaluation
- * UASG010 Quick Guide to Linkification
- * UASG018 Programming Languages Evaluation Criteria

Plus C-level outreach papers, magazine articles, presentations,

Who you are

This talk is a tutorial for those who know email addresses and Internet domain names primarily as ASCII-only. We introduce internationalised domain names (IDNA) and email addresses (EAI). Software development skills helpful for the exercises and some advanced material.

Primary audience

- * Users of domain names and email addresses, technically inquisitive
- * Application developers handling domain name and email addresses
- * Dev, QA, marketers, system administrators, and management

Context

The next 1,000,000,000 Internet users

Next 1 billion

China, India, Third World.
Large share use non-Latin script.
Little marginal North American,
European increase.
Mostly mobile and small-screen,
lower share on desktop, laptop.
Extending to mid-, lower-educated,
less comfortable with Latin script.

vs

First 1 billion

First world, N. America, Europe.
Large share use Latin script.
Includes large share of North
American, European potential.
Mostly desktop & laptop
computers, mobile only later.
Cream of highly-educated in each
market, the best at Latin script

Domain names

Domain names are the primary way to locate things on the internet. Original standards limited domain names an ASCII subset, and thus to Latin script. This obstructs users of non-Latin languages. They aren't just technical (see: ads), or written (see: saying a domain name)

Domain name standards

- * ASCII Letters, Digits, and Hyphen, max 63 (RFC1035)
- * Well known Top-Level Domains: .com, .org, .net, .jp, .ru, .cn, .in, ...
- * e.g. Amazon.com, XgenPlus.com,
- * Appear in many areas, e.g. email addresses, URLs, billboards, speech

Domain names, extended

Recent changes permit Internationalized Domain Names for Apps (IDNA). This allows new non-Latin TLDs, and non-Latin characters in rest of name. Parallel changes permit Latin TLDs with more than three characters. Thousands have been registered.

Domain name extensions

- * Internationalized Domain Names for Apps “IDNA2008” ([RFC5890](#))
 - * Replaces earlier IDNA2003
 - * e.g. `http:// 普遍接受 - 测试。世界`
- * `.भारत` (“bharat”, India), `.中国` (China), 「`。` 」 as well as `'`
- * `.tech`, `.museum`, and hundreds more

Email addresses

Still a mainstay of Internet communication. Actually a stack of related specifications, including SMTP, POP3, IMAP, *etc.* Original standards limited email addresses to an ASCII subset, and thus to Latin script. This obstructs users with names from non-Latin-script languages.

Email standards

- * Subset of ASCII, typically letters, digits, punctuation (RFC2822)
- * *mailbox @ domain.name*, e.g. info@unicode.org
- * *mailbox* preferably similar to user's own name in own script
- * Many implementations, some deviating from standards

Email addresses, extended

Domain name extensions brings change to the domain.name part of email addresses. Extensions to email address syntax permit almost any Unicode character in mailbox. Consequences ripple through SMTP, MIME, IMAP, POP3, and more.

Email Address Internationalization (EAI) standards

- * EAI Overview and Framework (RFC6530) + 6 more RFCs
- * EAI requires changes to several protocols and components
- * Change takes time, so EAI must interoperate with legacy email

So many top-level domain
names!

The older, simpler top-level domain names

The top-level domain name is the part after the final '.' Until 2001, there used to be a small set of 3-letter generic top-level domains, plus 2-letter country code top-level domains. They all consisted of latin letters.

Top-level domains, up to 2001

- * generic: com, edu, gov, mil, org
- * country-code, 2-letter: e.g. .ca, .uk, .eu
 - * Based on ISO 3166-1 standard, with supplements
- * Latin script, letters only

Exercise: top-level domains today

Resource

- * <http://data.iana.org/TLD/tlds-alpha-by-domain.txt>
- * Consider analysing with spreadsheet or Perl/Python code.

Questions:

- * How many top-level domain names (TLDs) now?
- * How many begin with “XN--” prefix? How many don’t?
- * What is the longest TLD not having “XN--” prefix?
- * How many 3-character TLDs are there now?
- * How many TLDs not having “XN--” prefix include digits or ‘-’?

Internationalized Domain Names for Applications (IDNA)

IDNA: Unicode names, LDH infrastructure

The Domain Name System was designed to permit only Letters, Digits, and Hyphens (LDH). It was reliable, but highly critical. When internationalising, rather than add more characters to the DNS, they mapped other Unicode characters to LDH.

`http:// 普遍接受 - 测试。世界`

Learn the IDNA reference knowledge

Learn about Internationalized Domain Names for Applications, understand Nameprep and Punycode, know when to use U-Labels and A-Labels. In due course, libraries should take over some of this for you.

IDNA references

- * [RFC5890](#) IDNA: Definitions and Document Framework
- * [RFC5891](#) IDNA: Protocol
- * [RFC5892](#) The Unicode Code Points and IDNA
- * [RFC3492](#) Punycode: A Bootstring encoding of Unicode for IDNA
- * etc....

IDNA U-Labels, A-Labels, NR-LDH labels

Domain names are separated by period '.' into *labels*. A label using anything outside Letters, Digits, and Hyphen (LDH) is a U-Label. There is a corresponding A-Label made of LDH. The IDNA algorithm converts between U-Labels and A-Labels. The familiar LDH labels are "NR-LDH".

DNS and IDNA "labels"

- * e.g. www.uasg.tech has three labels: "www", "uasg", and "tech"
- * LDH labels: must not start or end with "-", LDH only, max length 63
- * A-Labels: LDH labels, start with "xn--", valid Punycode output
- * U-Labels: Unicode string from reversing Punycode on A-Label
- * A-Label \leftarrow Punycode algorithm \rightarrow U-Label

Example U-Labels, A-Labels, NR-LDH labels

U-Label, A-Label pairs

- * 中国 ⇔ xn--fiqs8s
- * भारत ⇔ xn--h2brj9c
- * résumé ⇔ xn--rsum-bpad
- * après-ski ⇔ xn--aprs-ski-30a

NR-LDH labels

- * com, gov, ca
- * unicodeconference, iuc41
- * apres-ski

What are these?

- * munchen, münchen
- * museum
- * xn-trik-bpad, xn--trik-bpad

Try it!

- * <https://eai.xgenplus.com/Multilanguage-To-Punycode-Convertor.jsp>

Anatomy of an A-label



- * Basic code points are U+0000 to U+7FFF
- * If no basic code points in U-label, then no basic code points and no final hyphen in A-label

A-label references

- * [RFC3492](#) Punycode: A Bootstring encoding of Unicode for IDNA
- * etc....

Understanding the encoded deltas

The letters ending the A-label are a integers, LSB to MSB, digits a-z,0-9, self-delimiting. They encode both the Unicode scalar of a non-basic code point, and its location in the string. The details are complex.

Models of the encoded deltas

- * Deltas are tuples of (scalar, index into U-label)
 - * e.g. résumé \Leftrightarrow xn--rsum-[(U+00E9 'é',2), (U+00E9 'é',6)]
 - * \Rightarrow r_sum_ && [(U+00E9 'é',2), (U+00E9 'é',6)]
 - * \Rightarrow résum_ && [(U+00E9 'é',6)] \Rightarrow résume && [] \Rightarrow résumé

Understanding the encoded deltas

Refine the previous model by requiring characters in increasing code point order. And, store differences between scalars, which are smaller than the scalars. And store indexes into accumulated A-label, not the U-string.

Models of the encoded deltas

- * Deltas are tuples of (scalar-prev. scalar, index into A-label so far)
 - * e.g. résumé \Leftrightarrow xn--rsum-[(U+00E9-U+0080,1), (U+00E9-U+00E9,5)]
 - * \Rightarrow ${}_0r_1s_2u_3m_4 \ \&\& \ [(0x19,1), (0x0,5)] \Rightarrow$ ${}_0r_1\underline{é}s_3u_4m_5 \ \&\& \ [(0x0,5)]$
 - * \Rightarrow $\underline{r}\underline{é}\underline{s}\underline{u}\underline{m}\underline{é} \ \&\& \ [\] \Rightarrow$ résumé

Understanding the encoded deltas

Refine the model again by representing differences of scalars, and indices, as an integer (details complex). Represent the integer with digits 'a'-'z','0'-'9', where 'a'=0 and '9' is 35 or more (details complex). In LSB-MSB (reverse) order. A 'threshold' delimits MSB (details complex).

Models of the encoded deltas

- * Deltas are integers from (scalar-prev. scalar, index into A-label so far)
 - * e.g. (U+00E9-U+0080,1) \Rightarrow (0x19,1) $\Rightarrow 25*(4+1)+1 \Rightarrow 126 \Rightarrow$ 'bpa'
 - * and (U+00E9-U+00E9,5)] \Rightarrow (0x00,5) $\Rightarrow 0*(5+1)+5 \Rightarrow 5 \Rightarrow$ 'd'
 - * So xn--rsum-bpad \Rightarrow $_0r_1s_2u_3m_4$ && ['bpa', 'd']
 - * \Rightarrow $_0r_1\underline{e}_2s_3u_4m_5$ && ['d'] \Rightarrow resume && [] \Rightarrow résumé

Exercise: experiments with encoded deltas

Resource

- * <https://eai.xgenplus.com/Multilanguage-To-Punycode-Convertor.jsp>
- * <http://data.iana.org/TLD/tlds-alpha-by-domain.txt>

Questions:

- * Convert xn--rsum-bpad. Delete trailing 'd', then 'bpa'. Change 'd' to 'c'.
- * Convert xn--h2brj9c. Delete trailing '9c', then 'j', then 'r', then 'h2b'.
- * For each TLD with 'XN--' prefix, convert to Unicode. What are they?
- * Type various U-labels in Unicode box. Convert to A-labels.
- * Attempt to guess leading digits of deltas in A-labels (details complex).

Exercise: hand-run Punycode encoding

Resource

- * The simplified models of encoded deltas from previous slides

Questions:

- * For various U-labels, express as basic code points and:
 - * tuples of (scalar, index into U-label)
 - * tuples of (scalar-prev. scalar, index into A-label so far)
 - * integers from (scalar-prev. scalar, index into A-label so far)
 - * [Not really fair, details are complex, must refer to [RFC3492](#).]

Writing good apps in a world of Internationalized Domain Names for Applications (IDNA)

Tools & Resources for Developers

Authoritative Tables:

- * <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>
- * See SAC070 on static TLD / suffix lists: <https://tinyurl.com/sac070>

Internationalized Domain Names for Applications:

- * Tables: <https://tools.ietf.org/html/rfc5892>
- * Rationale: <https://tools.ietf.org/html/rfc5894>
- * Protocol: <https://tools.ietf.org/html/rfc5891>

Unicode:

- * Security Considerations: <http://unicode.org/reports/tr36/>
- * IDNA Compatibility Processing: <http://unicode.org/reports/tr46/>

Universal Acceptance
Steering Group info &
recent developments:
www.uasg.tech

Five Key Tasks of Universal Acceptance



Accept. Validate. Store. Process. Display. For all domain names.
Make wise end-to-end decisions about using A-Labels, U-Labels.

UASG guides

- * UASG006 Universal Acceptance Quick Guide

[http:// 普遍接受 - 测试. 世界](http://普遍接受-测试.世界)

Principles of Universal Acceptance



Accept

The process by which an email address or domain name is received as a string of characters from a user interface, file or API.

UASG Recommendations

- * User interface elements must support:
 - * Unicode.
 - * Strings up to 256 characters.
- * ASCII Compatible Encoded text ("Punycode") in place of Unicode.
 - * Unicode shown by default.
 - * Punycode text shown *only* when it provides a benefit.



Validate

UASG Recommendations

The process by which an email address or domain name – received or emitted – is checked for syntax correctness.

- * Easiest way to ensure all valid domain names are accepted.
- * Should not occur unless required. If yes:
 - * Verify TLD against authoritative table.
 - * Query domain name against DNS.
 - * Require repeated entry of email address.
 - * Validate characters - no “disallowed” code points.
 - * Limit to few, whole-label rules defined in RFCs
 - * If string contains '。' convert to '.'



Store

The long-term and / or transient storage of domain names and email addresses.

UASG Recommendations

- * Apps / services should support Unicode
- * Information stored in UTF-8 whenever possible
- * Consider end-to-end scenarios before converting between A-Labels & U-Labels
 - * Consider storing in both formats
- * Clearly mark email addresses and domain names during storage



Process

UASG Recommendations

Occurs whenever an email address or domain name is used by an application or service to perform an activity, or is transformed into an alternate format.

- * Check code points not defined when application / service was created – shouldn't "break" user experience.
- * Use supported Unicode-enabled APIs.
- * Use latest IDNA Protocol & Tables documents for Internationalized Domain Names.
- * Process in UTF-8 wherever possible.



Process (continued)

UASG Recommendations

- * Ensure numbers are handled as expected
- * Treat ASCII numerals & Asian ideographic number representations as numbers
- * Upgrade apps & servers/services together
- * Perform code reviews to avoid buffer overflow attacks

Occurs whenever an email address or domain name is used by an application or service to perform an activity, or is transformed into an alternate format.



Display

Display occurs whenever an email address or a domain name is rendered within a user interface.

UASG Recommendations

- * Display all Unicode code points supported by underlying operating system.
- * When developing app/service, or operating a registry, consider languages supported.
- * Convert non-Unicode data to Unicode before display.
- * End user should see “everyone. みんな” vs. “everyone.xn--q9jyb4c.”



Display (continued)

UASG Recommendations

Display occurs whenever an email address or a domain name is rendered within a user interface.

- * Display Unicode by default
- * Use Punycoded text *only* when it provides a benefit
- * Consider that mixed-script addresses will become more common
- * Use Unicode IDNA Compatibility Processing to match user expectations
- * Be aware of unassigned & disallowed characters

Linkification and Universal Acceptance

When you recognise URLs or IRIs and automatically make them links (“linkification”), do so with universal acceptance. If you have a detailed regular expression in your code, it is probably wrong. We have a guide.

UASG guides

- * UASG010 Quick Guide to Linkification
 - * Standards Principle: link all well-formed URLs
 - * Universal Acceptance Principle: treat all top-level domains & all scripts well
 - * Safe Practice Principle: various security considerations
 - * etc....

http: // اختبار-القبول العالمي.شبكة

Email Address Internationalization (EAI) issues

EAI Handling: your app + your stack

مانیش @ آشوکا. لهند
To:

To provide Universal Acceptance, your application must handle Email Addresses Internationalization correctly.

The various components which make up your email sending and receiving stack must also be support EAI.

Here is a quick guide to the high level issues.

To: données@fußballplatz.technology

EAI Universal Acceptance in your app

Email addresses can have international text in both the mailbox and domain name parts. If app tests email addresses with a detailed regular expression in your code, it is probably wrong. We have a guide.

UASG guides

- * UASG014 Quick Guide to Email Address Internationalization (EAI)
 - * Clients (Mail User Agent) display domain name in Unicode, send as A-Label; display and send mailbox name in Unicode
 - * Follow UASG010 Linkification guide for links in messages
 - * Consider validation via test emails rather than by address structure

EAI Universal Acceptance in your stack

Your stack of email sending and receiving components need to support EAI, and declare that they do. They may include SMTP, IMAP, POP3, and more. We have a guide and case studies.

UASG guides

- * UASG014 Quick Guide to Email Address Internationalization (EAI)
 - * Servers (Mail User Agent) advertise SMTPUTF8 support etc.
 - * Email service providers, consider ASCII alternative addresses, proper casing
 - * Transition challenges with non-EAI correspondents

Email Address Internationalization (EAI) resources

EAI case studies

We have case studies of organizations which have already supported EAI. Their experience helps you know what to expect. They may have tools you can use to help test your EAI.

UASG guides (partial)

- * UASG013D Case Study: Data Xgen Technologies Pvt Ltd
 - * “updating... at least 12 major elements... webmail, IMAP, POP, SMTP, contacts, calendar, antispam, search, logger and rules.”
- * UASG013C Case Study: ICANN
 - * Phased approach, 87 components = 46 in-house + 41 from vendors

UA use cases

IDNA Pattern	Example
ascii.long	ua-test.technology
idn.idn	普遍接受 - 测试 . 世界
idn-rtl.idn-rtl	اختبار - القبول للعالمي . شبكة
idn.ascii/unicode	普遍接受 - 测试 .top/ 我的页面
EAI Pattern	Example
unicode@idn.idn	युएअसजी@डेटामेल.भारत
ascii@ascii.idn	info4@ua-test 。 世界
unicode@rtl.rtl	دون@رسيل.السعودية

UASG004 Use Cases
These domains are
registered, ready to
use in test suites.
Total 45 cases.

XgenPlus tools

Software developer XgenPlus has made a number of EAI- and IDNA-related tools available to developers free of charge. Here are some links to start exploring.

XgenPlus tools (partial)

- * <https://eai.xgenplus.com/>

- * Puny Code Converter, Mix Script test, Mail Delivery Test

- * Datamail multilingual email service <https://www.datamail.in/>

- * Email addresses in 12 scripts for iOS, Android, and web.

Next steps

Learning more about IDNA and EAI

There is another session at IUC41 related to IDNA and EAI (by this same presenter!). The UASG stands ready to help support your use of IDNA and EAI, and to help you support others. Join us!

Suggested next steps for you

- * IUC41, Weds, session 10 track 1 “How does your framework rate?”
- * Follow the UASG at <https://uasg.tech/> .
 - * e.g. <https://uasg.tech/event/webinar-broccoli-issues/> , 19. Oct, 14h UTC-4
- * Subscribe to the ua-discuss@uasg.tech email list.
 - * <https://mm.icann.org/mailman/listinfo/ua-discuss>

Q&A

Thank you!

Q&A

Slides: <http://go.jdlh.com/iuc41t4t1>

Evaluation: <http://unicodeconference.org/eval-sp/>



Email addresses and domain names are
non-latin! Now what?

Jim DeLaHunt / IUC41 / 16 October 2017