Universal Acceptance
Association Newsletter Article 3

Programming Language Hacks

By Don Hollander
Secretary General
Universal Acceptance Steering Group

Universal Acceptance is the simple concept that all domain names and all 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 the mailbox portion of email addresses.

Examples taken from the UASG's working test cases include:

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 monthly.

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 that allows non-ASCII domain names (known as *U-labels*) to be represented in ASCII (known as *A-labels*.) No similar facility exists for mailbox names.

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

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

---

[1] Make sure there's one and only one '@', no consecutive dots '.', and the entire domain length is no more than 255 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

Universal Acceptance

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

Universal Acceptance