ICANN Internet Users Workshop
28 March 2006
Wellington, New Zealand

Who Said Anything About Punycode? I Just Want to Register an IDN.

Cary Karp
MuseDoma — dotMuseum

You don’t really have to know
anything more than the
name you want to register.

But there are a few additional things
that may be useful to keep in mind
before proceeding.

Computers are easily capable of dealing with a great many languages.

The typical working environment is, however, only configured to deal with the languages used in a single “locale”.

Keyboards, in particular, are highly language specific.

Available fonts may also be primarily intended for the display of characters most frequently needed in a given locale and have limited support for others.

An IDN that makes perfect sense to a user in one locale may only be of limited utility in another.

What happens when someone reading an IDN off your business card cannot type it on their own keyboard?

What happens when you are in an Internet café far away from home and find that the keyboard there doesn't have the characters you need to access your own site?

Locale sensitivity is not specific to IDN. Everyone is likely to have received e-mail that appears either partially or completely as a jumble of random symbols, rather than as plausible text in some language.

This is most likely because your computer does not recognize the way the characters are encoded or doesn’t have the resources needed for displaying them as intended.

There are many character encoding systems in current use. Most are intended to serve the purposes of a small number of languages.

These are simply lists that assign a unique number to every character that they contain.

Different encoding systems frequently use the same numbers for different characters. (That's what usually causes garbled e-mail and plenty of additional difficulty.)

There is one very large encoding system that accommodates a Universal Character Set.

This assigns “code points” to hundreds of thousands of characters and is still growing.

It is maintained in two coordinated international standards, of which the one is Unicode.

A request for the registration of an IDN will normally be made using the Unicode representation of that name.

There are several Unicode encoding formats. They are all displayed identically but use different numerical representations.

The one used for IDN is “UTF-8”.

Each TLD registry publishes a list of the characters that may be included in second-level domain names together with the corresponding Unicode code points.

If the name holder's computer is configured to use another encoding system, the conversion to UTF-8 will be made transparently at some point along the way.

What happens next is anything but transparent, although the requisite action does not usually need to be conducted by the name holder.

IDNs are not stored in the Domain Name System in their Unicode form.

The DNS uses another encoding system.

ASCII

American Standard Code
for Information Interchange

This permits the legible representation of domain names that contain characters taken from the 26-letter basic Latin alphabet,

a-z (both upper and lower case),

the ten digits 0-9,

and a handful of symbols, of which only the period and hyphen (“dot” and “dash”) are normally permitted in domain names.

All other Unicode characters need to be transformed into an

ASCII Compatible Encoding (ACE)

which is then used for the actual entry in the DNS.

A domain name label that is encoded in this manner is prefixed with

xn--

which indicates to a Web browser or other application software that the label needs to be decoded back into Unicode for proper display to the user.

There are several forms of ASCII compatible encoding and the xn-- prefix specifically designates a system called

Punycode

This is the only such scheme used for standards-based IDN.

The initial intention was that Punycode would never be exposed directly to users other than in situations where IDNs could not be displayed as Unicode characters.

In actual practice, such situations are numerous enough that the utility of IDN currently depends on some degree of user recognition and understanding of Punycode.

The basic structure of Punycode can be seen when it is applied to Latin letter that include diacritical marks.

The Punycode form of the label Nasenbär is

xn--nasenbr-bxa

The non-ASCII character is lifted out of the sequence and both its Unicode code point and position in the string are indicated in encoded form following the single hyphen. Note also that the upper case N is converted to lower case.

This is still apparent when additional decorated Latin characters are included:

næsebjørn

is

xn--nsebjrn-mxa0o

The shortest possible Punycode label is seven characters long even if it only corresponds to a single displayed Unicode character.

Punycode becomes totally cryptic when it represents non-Latin scripts:

коатимунди

is

xn--80ailbgohe1bm

Note also that the Cyrillic characters which look like the Latin o and a, as well as the lower-case letters that resemble upper-case Latin characters, are entirely distinct from them. This is the heart of the security concerns that attach to IDN.

Because of those concerns most Web browsers either require or permit special configuration for the selective display of the Unicode form of an IDN.

The Punycode form, although cryptic, is what is actually registered in the DNS and can thus be used in all applications, whether or not they are “IDN aware”.

There is an important further matter that needs to be considered with Punycode.

It is entirely possible that an encoded sequence that follows the xn-- prefix will be coincidentally meaningful in its own right.

For example, the string

xn--gibberish

actually decodes to a sequence of Arabic characters

ﺏ٩٧٨ﻓأإ

Although this particular string is meaningless, others may not be.

Because of all of these concerns, facilities for the interconversion of Unicode (UTF-8) and Punycode have become readily available on the Internet.

The full range of the underlying functionality can be tested at

http://josefsson.org/idn.php/

If the detail provided there appears daunting, lighter weight “UTF-8 IDN converters” are readily locatable.

There is an additional variety of utility software that makes the preparation and use of IDNs easier. This includes so-called “character pickers” that are used for the entry of Unicode characters that are not available directly via the keyboard.

(Something for consideration in a hands-on session at a future IDN workshop?)

Despite the relative ease with which a working environment can be configured to accommodate a wide range of characters, the typical user is set up for those associated with one locale only.

You need to take this into careful consideration if you expect your IDN usefully to cross locale boundaries.

The communication of an ASCII equivalent to the Unicode name is one obvious way to ensure its widest possible utility.

This can be done either through the parallel registration of a close ASCII equivalent to the IDN or by accepting the inelegance of the Punycode form and indicating it;

xn--gdb1cet1cq6b.tld may not be pretty

but it is robust.

Another option is to consider the linguistic contexts of each intended target audience and register IDNs appropriate to each.

This alternative will become of greater interest with the impending availability of IDN for top-level domain labels, which will also otherwise radically enhance the utility of IDN.

ICANN maintains a general list of IDN resources at:

http://icann.org/topics/idn/

Detailed information about security concerns that attach to the application of Unicode is at:

http://www.unicode.org/reports/tr36/

A further discussion some of the points raised in this presentation is included in an online tutorial at:

http://about.museum/idn/tutorial.html

Questions?

ck@nic.museum