

Designing, Building & Operating an IDN ccTLD Registry

Chris Wright

CTO - AusRegistry International

ICANN no. 35, Sydney, Australia

24th June 2009

AusRegistry International

- Located in Melbourne, Australia
 - Involved in Domain Name Industry since 1999
 - ICANN Accredited Registrar since 2000
 - .au Registry Operator since 2002
- Domain Name Registry Services
 - Registry Systems and Software Provider
 - Consultancy Services
 - Our software and consultancy services have been used by several other TLDs including some IDN enabled ccTLDs

Overview

- Implementing IDN representation of a ccTLD
- Implementing IDNs in an ASCII ccTLD
- Overview of some of the issues when designing, building and operating an IDN enabled Registry platform
- Not a technical presentation

Agenda

- Are you ready?
- Levels of sophistication
- IDN specific policy
- Internationalising your whole Registry platform
- Other considerations
 - Similar registrations
 - Bundles
- Implications
 - DNS
 - Registry performance
 - Registrars, Registrants and end users
 - Entire platform considerations

Are you ready for IDNs?

- IDN ccTLD fast track is just around the corner
- IDN new gTLDs are coming

IDN enabling a Registry Platform (or building one from scratch)

- Very Very Simple Implementation
 - Just accept xn-- registrations in your existing ASCII Registry system
- Very Simple Implementation
 - Accept xn--
 - Also request a language tag and store it
- Simple Implementation
 - Accept xn--
 - Request a language tag and ensure it is protocol valid
 - Verify that the Unicode code points it maps to are in a restricted set that applies to the language in use
 - All other operations work on xn-- form only

Is a naive solution really what we want for national infrastructure?

As responsible ccTLD managers we must...

- Minimise public, Registrant and Registrar confusion
- Protect against phishing and other misdirection style attacks
- Maintain
 - high security standards
 - high performance standards
 - policy rich controls (where relevant)
- Protect the reputation of our namespace
- Manage our National infrastructure the way an important international asset should be managed

Given the previous points, the naive solution is just not the right way to go for most of us

With that in mind...

- Some of the more important aspects to consider in a more responsible implementation of IDNs include:
 - Developing IDN specific policy
 - Fully Internationalising your Registry Platform
 - Blocking similar registrations
 - Bundles
 - Variants of your IDN zone
 - Effects on DNS
 - Security considerations
 - Performance impacts
 - Effects on Registrars, Registrants and end users
 - Implications for Registry Website & other interfaces

Developing IDN Specific Policy

Don't skip this step

- It is important to understand what rules you want to implement:
 - IDNA only describes:
 - Allowable Unicode code points
 - Broad validity rules
 - BIDI rules for string stability
 - Normalisation rules (NFC)
 - Left to local policy:
 - Unicode code points that make up your language(s)
 - What code points or sequence of code points you consider to be variants of one another
 - What mappings (if any) you will expect application developers / registrars to apply
- All must defined and technically enforced without compromising the integrity of your Registry system

Fully Internationalising your Registry Platform

Fully Internationalising your Registry Platform

- Further work
 - Fields :
 - Contact properties
 - Domain passwords (authinfo in EPP terms)
 - Host names
 - Email addresses
 - Interfaces :
 - WHOIS
 - Registry website
 - Invoices & accounting statements
 - Help desk tools
 - Monthly reporting

Blocking Similar Registrations

What do we mean by blocking similar registrations?

- In ASCII domain names we do this implicitly e.g.
 - example.com
 - Example.com
 - EXAMPLE.com
 - ExAmPIE.com
- In this particular case this is enforced by the DNS protocol

However with IDNs...

- There are many more cases where blocking of registrations may be required e.g.

Convention, visually confusing or historic	Non-visual reasons	Technical reasons
café.com cafe.com	۱۱۱۱۱.com 11111.com	ا.كوم (U+0627,U+0654) أ.كوم (U+0623)

- No single, simple rule can be applied, i.e. just lower casing does not help

Sometimes blocking just isn't enough!

- Example: café.com
- The registrant may not only expect cafe.com to be blocked but may also want cafe.com to function
- A single domain name registration may need to result in multiple entries placed in the DNS
- Commonly referred to as bundling

There are better ways...

- Context or reason for the variant
 - The domain 11111111.com should only have the following variants:
 - 11111111.com
 - 11111111.com
 - Now only two combinations
 - And even if a domain name had 32 characters there would still be only two variants

Other bundling considerations

- Allowing Registrants to turn parts of a bundle off or on
 - How?
- Impacts on other services offered
 - e.g. DNSSEC
- Charging model
 - Should there be one?
- Flow on effects to accounting and reporting
 - Is a bundle of three domains one registration or three?

Variants of your IDN zone

What if your zone was .café?

- An example:
If George registered a domain name for the cafe Take Away Cafe he might register the following
 - `takeaway.café`
- He may also reasonably expect to have the domain
 - `takeaway.cafe`
- Implementing this should be simple, and is important to consider
 - Those still generating zone files should be able to use the `$ORIGIN` statement
 - Those using dynamic updates or other methods will need to take more care, however it can still be done
 - It does raise many new issues though, about keeping multiple TLD zone files in sync (what if an update to one zone fails?)

Effects on DNS

The obvious effects

- If allowing bundles, an increase in the size of the zone file
 - 100,000 names each with 10 variants = 1,000,000 domains
 - If each has 2 name servers = 2,000,000 entries
 - Without bundling this would only be 200,000 entries
 - That is a 900% increase in size!
- If you outsource your DNS that could be a significant cost
- Managing large zone-files with traditional tools is not easy
- Managing IDN zone-files with traditional tools is not easy
 - xn--e1afmkfd. xn--80akhbyknj4f <- - - - What domain is this?
- Zone generations techniques may need to be changed to still meet SLTs

The not-so obvious effects

- If allowing bundles, how should we deal with DNSSEC?
 - Whilst we can use the same name servers we can't use the same DS records

- DNAME vs. Delegations

	DNAME	Delegations
Pros	<ul style="list-style-type: none"> •Really easy to use for Registry & Registrant 	<ul style="list-style-type: none"> •Easy to understand •No loss of functionality
Cons	<ul style="list-style-type: none"> •No ability to have records at the domain name itself (no EMAIL!) 	<ul style="list-style-type: none"> •Registrant has to setup and maintain multiple zone files (can use BIND tricks) •Lots of records to be kept consistent

- DNAMEs and DNSSEC

Security Considerations

New attack vectors

- IDNs are significantly different, even simple mistakes can cause problems if not anticipated
- Punycode overloading
 - Specially created punycode stings can be made to easily generate code point numbers beyond Unicode (invalid or unassigned)
 - If you implementation doesn't detect and handle these conditions you are asking for trouble
- Punycode reverse engineering
 - xn--trademark
 - xn--rude-or-offensive-word
- Variant overflow (deliberate or accidental)
 - 2^{59} is a really big number!
- Phishing and other scams
- Supplementary Characters
 - libIDN a popular Java implementation of IDNs DOES NOT handle these correctly
 - Any Java 1.4 libraries that haven't been updated to use new java functionality will fail on these (any new implementations not using new functions will also fail)

Performance Impacts

Not being naive has its downsides...

- Many ccTLD Registries include performance and SLTs
- Validation rules and cross checking that now needs to be performed has to be implemented as **streamlined** as possible, especially doing this on domain availability checks
- A lot of ASCII 'tricks' or optimisations are invalidated e.g.
 - Byte size != string length
 - Byte equivalency is not the only case of equality any more
 - Lower casing is not the only pre-processing required for uniqueness checks
- Multi-zone registries with mixed IDN and non-IDN zones will even incur a performance hit on the non-IDN enabled zones as certain checks still need to be performed

Effects on Registrars, Registrants and End Users

Effects on Registrars, Registrants and End Users

- It is different to ASCII domains
- Registrars have a harder job to do now
 - Interpret what the Registrant wants
 - Turn it into something remotely protocol valid (to map or not to map?)
 - Explain all of this to the Registrant
- Provide tools to Registrars
- Ensure consistent message to Registrants and end users

Registry - Registrar Protocol

- EPP doesn't realistically natively support IDNs
- ICANN guidelines request at least a language tag is supplied with each IDN domain name
- This tag is required to do server side processing optimisations
- Which form of the domain should you accept?
 - DNS Form: xn--dabnbdasf.com
 - User From: *<insert Arabic here>*
 - Both? Either?
 - IDN protocol recommends you should get both, EPP doesn't allow this
- User Friendly Error Messages
- Management of Bundles

Implications for Registry Website & Other Interfaces

All those things Registrars had to do...

- Registry 'Human' interfaces also have to be updated
 - Apply Local Mappings, etc.
- Interfaces include
 - WHOIS (web based and port 43)
 - xn-- and user form?
 - Unicode in response (UTF8?)
 - Registry website
 - Searching and other Lookup functions
 - Interface itself probably should be multi-lingual
 - Help Files
 - Registry EPP Services
 - Poll Messages
 - Error Messages
 - Accounting and Reporting Interfaces
 - Zone Tools

In Summary

Doing it 'right' is hard...

- ... but, at least in our opinion, necessary
- Implementing IDNs in a policy rich environment whilst maintaining high standards of security, performance and flexibility is difficult
- But not impossible
- Don't give in and take the easy way out, it will prove harder in the long run
- There is help out there if you need it!

