Tagged non-US-ASCII character sets in DNS labels

Rob Austein, InternetShare.com
sra@hactrn.net

Two separate questions

Assume that we must add international charset support to the naming system somehow

#1: Should we put it in the DNS or someplace else?

#2: If we're putting it into the DNS, how shall we do it?

Question #1 is the important one

The light is better at question #2

Should we put it in the DNS?

Pros:

- DNS is what we have and what the world knows
- May already be too late to stop this
- "Just send 8" in use already in several places
- Can we re-use existing battleground for next round of name wars?
  - Rather than creating a whole new battleground
- Adding non-US-ASCII support to DNS itself isn't very hard

Cons:

- DNS is already stressed from being used for things it's not particularly good at (eg, directory services). Perhaps we should leave the poor thing alone and Internationalize something else
- Normalization is hard, and none of the solutions are satisfactory
  - How complicated and slow are we willing to make this?
  - Think about typing in an email address from a business card
- Adding non-US-ASCII support to DNS is the least of our problems. The hard one is adding it to all the applications

The Choices

a) Stick with US-ASCII, address the problem elsewhere

b) Transition DNS to DNS + Unicode

c) Transition DNS to MIME model

- Support "all" charsets
- Tag the charsets so that we can figure out what we're looking at

Implications of chosing US-ASCII

Some people will do "just send 8" anyway

"Just send 8" communities won't interoperate except by luck

Problem will pass beyond IETF control

Implications of chosing Unicode

UTF-8 is probably best choice if we go to Unicode

Expect minor problems with deployed DNS protocol code

Expect non-trivial problems with applications

Will the world accept Unicode?

- Opinions vary
- How many Unicode MIME messages have you received to date?

Implications of chosing tagging

Relatively minor change to current DNS protocols

Normalization is hard, but:

- Normalization for "just send 8" is effectively impossible
- Normalization isn't easy even within Unicode

Require upgrade of all DNS software

Tagged charsets in DNS

Remember, we haven't decided to do this at all yet

But if we do, let's learn from the MIME folks

There's no consensus on character sets

- Some say Unicode
- Some prefer ISO-8859-*, JIS*, ...
- Problem is scheduled to be solved immediately after cure for cancer

So perhaps we should get a tagging framework in place before "just send 8" becomes the norm

קאקאמיימי.example

"קאקאמיימי" label

Octet 0: Code indicating "labeled charset label"
RFC-2671 3.1 "extended label" code XXX [TBD]
Octet 1: Total label length, including octets 0-5
In this case length is 15
Octets 2-5: IANA charset registry MIBenum code, big-endian
In this case, code 11 for ISO_8859-8:1988
N.B.: No new IANA registry required!
Octets 6-N: Label text encoded in specified charset
In this case: 247 224 247 224 238 233 233 238 233

"example" label

Normal US-ASCII label

Invisible root label

Normal zero length root label

Putting it all together

XXX 15 0 0 0 11 247 224 247 224 238 233 233 238 233
7 101 120 97 109 112 108 101
0

Normalization

Problem: mapping glyphs to codes

- In what order does resolver try multiple charsets?

Partial solution: new RR type specifying order

- Call this the "CSNY" RR type for now
  - "Character Set Normalization, Yecch"
  - Type code and real name TBD
- Does not address intra-charset normalization, eg within Unicode

קאקאמיימי.example. IN CSNY US-ASCII ISO-8859-8

For backwards compatability, keep US-ASCII at front of list

Additional section processing / glue RR problem here

How much does this slow down queries?

Don't forget the real question

Perhaps this stuff just doesn't belong in DNS

Deciding "how" is premature at this stage