Internationalization Update
Revising IDNs

Identification of Perceived Issues

- Review and Recommendations for Internationalized Domain Names
  - Covered perceived issues in the community
  - No recommendations about actions other than need to do some updating
- Published as RFC 4690, September 2006

New Proposals from Small Design Team

Effort to…

- Implement “inclusion” model
- Correct parts of IDNA that make some languages impossible
- Reject non-language characters
- In general, eliminate remappings in favor of prohibition of characters that are mapped out
- Recast IDNA model somewhat into procedural terms, not “implement this algorithm”
- Adapt to new Unicode versions

Revised View of IDNA Registration I

- Start with proposed label
- Convert from local environment or conventions to Unicode if needed
- Identify permitted chars, reject labels with others. Preprocess if needed.
- Map through stringprep

Revised View of IDNA Registration II

- Postprocess if needed
- Apply registry restrictions and language checks
- Convert to punycode
- Insert in zone

IDNAbis Lookup

- User Input
- Conversion to Unicode
- Validation and preprocessing
- Stringprep
- Postprocessing
- Punycode
- Name resolution

Changes, Compatibility, and Prefixes

- Avoid changes in interpretation
- Old valid string should yield either
  - Same punycode or Invalid
  - ToUnicode(ToASCII(ToUnicode(string))) is stable
- Old invalid string might be valid
- Back translations from valid punycode yield same results as before

Still Some Major Issues With

- Ligatures and Digraphs
  - Cannot really be resolved properly with available info
  - “Presentation forms”
- Implausible ideas
  - Put any word (or phrase?) in any language into DNS
- adn…

IDNA and Right-To-Left

- IDNA permits domain names that mix RTL and LTR labels
- IDNA restricts RTL labels
  - Any label must be fully RTL; no mixing
  - Last and first character must both be definitely RTL, because context is unknown
  - Middle characters can have ”neutral” direction; they are made RTL by the end characters

Major constraint

- Labels ending with combining marks cannot be used
- They end in a character that is of ”neutral direction”, and IDNA forbids them
- Any language in which such marks are obligatory will therefore not be available

Known problematic cases

- Dhivehi, the national language of Maldives, is written in Thaana script with a combining mark on every base letter
  - ކޮމްޕިއުޓަރު(computer)
- Yiddish is normally written with combining marks (unlike Hebrew), that can appear in label-final position
  - ייִוואָ (YIVO)

One possible solution

- Allow for ”neutral direction” characters at the end of a word, as long as the UAX#9 rules make it RTL, even when followed by an LTR character
- Easy for the case of combining characters with neutral direction
- Other cases, including numbers, need to be considered carefully, and may stay as ”not permitted”

Challenges

- Define precisely the invariant properties we want RTL labels to have
  - Stay together in all contexts
  - Can be displayed consistently
- Create minimally restrictive rules for RTL labels that preserve the properties
  - Formulated in terms of UAX#9 rules
  - Make sense in as many linguistic contexts as possible

Tables and Mappings

- Specific to Needs of the DNS and IDNs
- Most normalizations & compatibility…
  - Treated as a Localization, OS, or UI issue
  - Must occur pre-IDNA
  - Other approaches lead to madness given language concerns

The Unicode Versioning Problem

- **Apps may not know version**
  - Need procedures that are stable with version changes
  - Can’t rely on promises: Unicore will correct problems (and maybe should)
- **Can determine which scripts and characters are permitted at a given time**
  - Requires slightly different “store” and “lookup” models with different verification details
  - Just like DNS, etc., etc.

Next??

- Work on new stringprep definition with
  - Explicit dependencies on Stable NFKC
  - Less mapping work due to exclusions
  - Complete stability for Stringprep2003-valid strings that remain valid
- Try to resolve differences with Unicore
- Figure out how to review, revise, and adopt

Summary

- This is a tuning/updating process, not a radical conceptual change
- It is necessary for Unicode version evolution and will help with “confusion” problems
- It is forward-compatible with existing IDN labels that use language chars, but not with all possible presentation forms

Reading Material

- Base: RFC 4690
- Design Team IDNAbis proposals
  - draft-klensin-idnabis-issues-00
  - draft-alvestrand-idnabis-bidi-00
  - draft-faltstrom-idnabis-tables-01
    - http:/stupid.domain.name/idnabis/draft-faltstrom-idnabis-tables-00.html
- Mailing list: idn-update@alvestrand.no