



IDNA - what's up?

Patrik Fältström
paf@cisco.com

In the beginning

- 3454 Preparation of Internationalized Strings ("stringprep"). P.
Hoffman, M. Blanchet. December 2002. (Format: TXT=138684 bytes)
(Status: PROPOSED STANDARD)
- 3490 Internationalizing Domain Names in Applications (IDNA). P.
Faltstrom, P. Hoffman, A. Costello. March 2003. (Format: TXT=51943
bytes) (Status: PROPOSED STANDARD)
- 3491 Nameprep: A Stringprep Profile for Internationalized Domain Names
(IDN). P. Hoffman, M. Blanchet. March 2003. (Format: TXT=10316 bytes)
(Status: PROPOSED STANDARD)
- 3492 Punycode: A Bootstring encoding of Unicode for Internationalized
Domain Names in Applications (IDNA). A. Costello. March 2003.
(Format: TXT=67439 bytes) (Status: PROPOSED STANDARD)

What is this?

- 3454 Specifies overall algorithm - stringprep
- 3490 Specifies IDN algorithm - IDNA
- 3491 Specifies Nameprep
- 3492 Specifies Punycode

stringprep

- With profiles, any Unicode based string can be converted to another Unicode string so that they can be compared
 - Include illegal codepoints
 - Include mapping table
 - Give ability to create profiles
- Used for IDN, LDAP and other protocols

idna

- Algorithm for how to convert a domain name with Unicode codepoints to ascii
- How to use the stringprep profile and unicode
- Includes specification on how to handle unallocated codepoints
- “core” to IDN standard

nameprep

- Specific stringprep profile for unicode based domain names
- Convert a domain name with unicode codepoints to one of
 - Illegal domain name
 - Domain name with Unicode codepoints

punycode

- Converts a label with unicode codepoints to a domain name in ascii
- Example:
 - fältström
 - xn--fltstrm-5wa1o

What happened?

4690 Review and Recommendations for Internationalized Domain Names
(IDNs). J. Klensin, P. Faltstrom, C. Karp, IAB. September 2006.
(Format: TXT=100929 bytes) (Status: INFORMATIONAL)

In short...

- Explains the problems in the earlier standards
 - Bidirectional scripts
 - Non-spacing codepoints
- Explains the problems with scripts not yet created when IDNA was written
- Explains problem with versioning of Unicode
 - Old standard based on Unicode 3.2

Example

- If a label include a character that has right to left directionality, both first and last character of the string has to have right to left directionality
- Creates problem if for example the string ends with a codepoint with no directionality

יִי וּוּאָ

U+05D9 HEBREW LETTER YOD (R)

U+05D9 HEBREW LETTER YOD (R)

U+05B4 HEBREW POINT HIRIQ (NSM)

U+05D5 HEBREW LETTER VAV (R)

U+05D5 HEBREW LETTER VAV (R)

U+05D0 HEBREW LETTER ALEF (R)

U+05B8 HEBREW POINT QAMATS (NSM)

- Note that last codepoint has no directionality (Non Spacing Mark)

יִיִוֹא
T

U+05D9 HEBREW LETTER YOD (R)

U+05D9 HEBREW LETTER YOD (R)

U+05B4 HEBREW POINT HIRIQ (NSM)

U+05D5 HEBREW LETTER VAV (R)

U+05D5 HEBREW LETTER VAV (R)

U+05D0 HEBREW LETTER ALEF (R)

U+05B8 HEBREW POINT QAMATS (NSM)

- Note that last codepoint has no directionality (Non Spacing Mark)

New IDNA standard

- Will consist of a few documents
- Will not change punycode
- Backward compatible
- Does explicitly talk about what is possible to register in the DNS (old IDNA say what is possible “*to use*”, and that include mappings
- Today's drafts **ONLY** define what is possible to register in DNS (U-label / A-label)

New documents

Current versions

- [draft-klensin-idnabis-issues-07](#)
- [draft-klensin-idnabis-protocol-04](#)
- [draft-alvestrand-idna-bidi-04](#)
- [draft-faltstrom-idnabis-tables-05](#)

New documents

New versions

- draft-ietf-idnabis-issues-00
- draft-ietf-idnabis-protocol-00
- draft-ietf-idnabis-bidi-00
- draft-ietf-idnabis-tables-00

draft-ietf-idnabis-issues

- In fact named “Rationale and issues...”
- Addresses the concerns in the IAB document RFC 4690
- Explain how the issues are resolved

draft-ietf-idnabis-protocol

- Replaces the IDNA specification
- Core specification of new IDN standard

draft-ietf-idnabis-bidi

- Gives specifics for bidirectional scripts

draft-ietf-idnabis-tables

- Defines algorithm to use to calculate whether a codepoint in Unicode is in one of the categories
 - PVALID (Protocol Valid)
 - CONTEXTO / CONTEXTJ
 - DISALLOWED
 - UNASSIGNED

But IDNA2003 had mappings

- Mappings are not part of IDNA2008
- Labels **MUST** be stable under NFC
- Codepoints in label **MUST** pass bidi requirements
- Codepoints **MUST** be ok according to algorithm specified in tables document (which might include contextual rules)
- We **MIGHT** see a separate document on mapping, recommended behaviour for different applications etc

Implications for registry

- Should not require any change for a registry that do have a policy for what subset of IDNA2003 codepoints one can register
- Changes might be required in two cases:
 - ➔ Registry that allow **any** IDNA2003 codepoint
 - ➔ Registry that define codepoints based on possible **input** to IDNA2003 (and not what is actually registered in DNS)

Why is this needed?

- IDNA standard must be independent of Unicode version
- IDNA standard must handle bidirectional scripts
- ...plus other things mentioned in RFC 4690

When will it be ready?

- “With in 6 months”
- Seriously: Request to people to write code based on the new standards. Last round of very careful review. Should go to official IETF review process during 2008. BOF at IETF in Philadelphia (March 10-14).
- Now a working group!
- Mailing list: idna-update@alvestrand.no

Patrik Fältström

paf@cisco.com