IDN and applications

Michel Suignard
Senior Program Manager
Microsoft
IDN is the first step

- IDN solves a DNS limitation by carrying extended domain entities within the existing framework
- But most users interact with resources, not host names: IRI anyone?
- Resource naming policies
- Legacy support
- Security
Resource Identification today is:

- URI (universal Resource Identifier)
  - ASCII only
  - Weak escaping mechanism (No or limited escaping reversibility)
  - No full interoperability for charset escaping
- De facto Internationalized URI
  - Non IDN conformant (lack of filters, case folding not addressed)
  - Bidirectional rules not addressed
  - No rules for conversion between ASCII and larger repertoires
Internationalized Resource Identifiers (IRI)

- Specifies internationalized protocol element
- Covers character encoded and un-encoded scenarios (side of the Bus case, movie credit, etc…)
- Fully specified mapping to URI
- Support Bidirectional (Hebrew-Arabic) scenarios
- Anchored on Unicode 4.0 / ISO/IEC 10646:2003
- Unicode Normalized (more for host)
- Related to IDN through the ‘authority’ component

http://讀賣新聞.co.jp/日本語/
IRI usage

- Existing schemes (http, ftp, mailto) should not use it directly
- Protocol element for new protocol or presentation element for presentation layer of existing protocols
- De facto usage in Browser address bars (URI presentation layer)
- Already implied by many XML languages/protocols (anyURI schema type)
Usage scenario for http

http://読売新聞.co.jp/日本語/

http://%E8%AE%AC%E8%B3%A3%E6%96%B0%E8%95%85%E8%81%9E.com/%E6%97%A5%E6%9C%AC%E8%AA%9E/

http://xn--efvv70di1hulb.com/%E6%97%A5%E6%9C%AC%E8%AA%9E/

host = xn--efvv70di1hulb
Bidirectional IRIs

- Use logical order (not visual order)
- Presented as if embedded Left to Right
  - Avoid reordering interaction with characters preceding and following the IRI
- Restrictions on host names:
  - Label cannot use both RtL and LtR characters,
  - Label using RtL Characters must start and end with them.
- Same restrictions should be applied to other IRI components, exceptions:
  - Opaque IRIs (never seen by users)
  - Query components (may be free format)
Bidirectional examples

http://سلام.دائم/path?query

1 2 3

http://123/سالم.دائم?query

1 2 3

http://سلام.دائم/123?query

1 2
Bidirectional examples (continued)

http://سلام.دائم/۲۳۲؟معکم

1 ← 2

http://سلام.abc.دائم/۲۳۲؟معکم

1 ← 2 ← 3 ← 4

http://سلام/دائم/سلام/Path-part/۲۳۲؟معکم

1 ← 2 ← 3 ← 4
Resource naming policies

- Internationalized host names should obey a language based name policy
  - i.e. A Polish name is not supposed to contain Arabic characters or even some other Latin based characters
  - Can be enforced by NICs, not necessarily by software
  - Existing rules for CJK characters: RFC 3743
  - May not be enforced/enforceable in sub-zones

- Multi-script registration should be rare, especially among Latin, Greek and Cyrillic
Legacy

- For good or bad reasons there is a large body of non ASCII DNS deployment in private Internet networks
- UTF-8 is widely used in these private networks
- Collision with the Punycode rule, especially when the internal DNS structure is mirrored in the public DNS structure (common scenario)

山田.株式会社.co.jp
å±±ç”° . æ a¼ å¼šç¾ . co.jp
xn--rhtu98c.xn--6oqv20b1zgzxr.co.jp
å±±ç”° .xn--6oqv20b1zgzxr.co.jp
Security issues

Which one is the resource locator I trust?

http://한국일보/사회면 or http://xn--3e0bm80ac2h5no/%EC%82%AC%ED%9A%8C%EB%A9%B4 ？


- Within these two groups all names locate the same resource
- For security reason it is important to pick one as the canonical representation
- If the Unicode name is selected, it **must** be normalized
Name spoofing

- Not a new concern
  - already exist in ASCII with 0, 0, 1, 1
- Much worse with Unicode repertoire
  - Cyrillic ‘Latin’ look alike: ABCEHIJKMOPY
  - Greek ‘Latin’ look alike: ABEHIKMNOPRTXYZ
  - Cherokee ‘Latin’ look alike: ABCEGHJKLMPRSTYVWZ
- Cannot be fully solved by restricting to a single script/language
- Identity crisis: how do I know who you really are?

www.example.org

is in fact:

www.xn--ml-6kctd8d6a.org
User Interface limitations

- Ubiquity versus Market adaptation
  - ASCII digits and letters have widespread adoption (example: phone number)
  - Market customization creates solutions that are opaque to most
- Often difficult to display and enter resource identifiers outside of the customer language usage area

한국일보/사회면

讀賣新聞.co.jp/日本語/
What applications/middleware can do?

- Implement IRI now
- Validate IRI and Punycode host names early on
- Consistent rules about Punycode and native Unicode display
  - Favor Unicode display
  - Discourage Punycode value direct input
- Provide display capability for all IDNA character repertoire
- Enforce IDNA and IRI Bidirectional string rules
- Help users determine resource identity
  - Language, script filters
  - Do not try to resolve ill formed host names
- Make trustworthiness the highest priority
Where is Microsoft?

- IDNA basic functions implemented in the next .Net Framework release (code Whidbey)
  - System.Globalization.IdnMapping class
  - GetUnicode and GetAscii members provide host name conversion between native Unicode and Punycode
- Equivalent native versions (Win32) planned for next version of Windows:
  - ASCIlToIDN()
  - IDNToASCII()
Internet Explorer status

- Its presentation layer already uses IRI (address bar, status bar)
- Its URI layer needs to be updated to map IRI to URI according to the IRI specification
- Non ASCII host names still need to be converted according to IDNA specification
- Some remaining issues:
  - What to do with illegal IRIs/host names?
  - UTF-8 legacy
  - Proxy protocol (UTF-8 or Punycode)
  - Security impact
Questions?