PRECIS and i18n

IETF Tutorial
IETF 94 @ Yokohama, Japan
A brief history
A brief history

• The beginning of the Internet was ASCII only
• It was enough for the researchers

• The Internet growth was accelerated since late 1980s, due to popularization of Work Stations (WS)
• And the user of the Internet was expanded to engineers, designers, etc.

• ASCII was not enough for their communication
A brief history

• In early 1990s, MIME (Multipurpose Internet Mail Extensions) was standardized

• Various character sets (other than ASCII) were enabled to use in e-mail message body (and some unstructured headers)

• Of course, in hyper-text messages also

• In middle 1990s, appearance of commercial ISPs and the Internet capable consumer OSs changed the Internet users drastically

• The door was opened for “end users”
A brief history

• The Internet were spread all over the world
• E-mail and WWW became the end users’ daily communication tool
• They used their native language and scripts on the Internet

• In late 1990s, some portion of the end users were desired to use their internet identifiers in their native language
• That were, Web addresses and e-mail addresses
Internationalization in the IETF
RFC 2277

- Aka BCP 18

- It requires “internationalization considerations”
- Also suggests “just use UTF-8”

- We know, the first was widely ignored, and the second is not enough
i18n

• internationalization
• 18 characters between ‘I’ and ‘N’

• Simiraly
• l10n for “localization”
• m18n for “multilingualization”
i18n in the IETF

• Some people say that the Internet is used international, so it is internationalized from the beginning (→ globalization)
• Some people say it is localization (→ localization of communication)

• In the IETF, "internationalization" means to add or improve the handling of non-ASCII text in a protocol [RFC 6365]

• i18n provides common framework to communicate in local text globally
i18n of identifiers in the IETF
IDN

• Internationalized Domain Name
• The first fundamental i18n work of identifier in the IETF

• IDNA2003
  • RFC 3490, 3491, 3492

• stringprep
  • RFC 3454

• IDNA2008 (Obsoletes IDNA2003)
  • RFC 5890, 5891, 5892, 5893, 5894
IRI

• Internationalized Resource Identifier
  • RFC 3987

• Update work is in progress at W3C
EAI

• Email Address Internationalization

• Experimental
  • RFC 4952, 5335, 5336, 5337, 5504, 5721, 5738, 5825, 5983

• Standard
  • RFC 6530, 6531, 6532, 6533, 6855, 6856, 6857, 6858
Backward compatible or not?

• IDN has backward compatibility with ASCII infrastructure

• EAI has (almost) no backward compatibility with ASCII infrastructure

• Balance between pros and cons
  • Deployment
  • Easiness of implementation

• Choice of protocol design
Issues in i18n of identifiers
Unicode

• A character set
• Consists from various scripts, punctuations and symbols used in world wide
• More than 100K characters
• Now it has Emoji 😊
• The notation ‘U+xxxx’ stands for Unicode code point

• UTF-8 [RFC 3629] is an encoding method of Unicode code points
Why not “just use UTF-8”?

• It is OK in “contents”
• The end users can recognize what is displayed
• Of course, must pay attention for phishing

• It is not OK in “identifiers”
• Sometimes, a displayed character has several computer/network internal representations (code points / sequence of code points)
• Identifiers need to be match exactly
• What happened if they are the same for human eyes but differ for devices?
  • für (U+0066 U+00FC U+0072)
  • für (U+0066 U+0075 U+0308 U+0072)
What is “the same”?

• Case
  • $a \leftrightarrow A$
  • $i \leftrightarrow \breve{i}$ (language dependent)
  • $\breve{f} \rightarrow SS \leftrightarrow ss$ (no round-trip)

• Mapping
  • ZWJ $\rightarrow$ nothing (contextual)

• Equivalence (Normalization)
  • $\ddot{u} \leftrightarrow u + \breve{u}$ (canonical)
  • $\breve{a} \leftrightarrow \breve{p}$ (compatibility [kompatibility])

• “The same” is not common between protocols
stringprep

• Is a deliverable of IDNA2003
• It defines framework for preparing identifier strings to compare

• Profiles to select options
  • Mapping
    • Case
    • To nothing
  • Normalization
  • Prohibit check
    • Such as control characters
  • Bidi check
    • c.f. next page
Bidi

• Bidirectional
• Some scripts such as Arabic have property to write right-to-left
• Mixed use of right-to-left and left-to-write scripts cause huge confusion

اللغة العربية → اللغة العربية
Is stringprep perfect?

• Unfortunately, not

• Case mapping and Normalization are depends on the Unicode specification

• Those specifications are updated when the Unicode version is updated

• stringprep strictly depends on the Unicode version 3.2.0

• The most recent version of the Unicode is 8.0
Is stringprep perfect?

• Normalization option is restricted to canonical compatibility (NFKC) or nothing

• Operational experiences of IDNA2003 revealed that canonical normalization (NFC) is preferable for IDN

• IDNA2008 addressed those issues, regardless of stringprep
Is stringprep updated?

• Yes!

• That is, PRECIS
PRECIS
What is PRECIS?

• Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols

• RFC 7564
What is new in PRECIS?

• Unicode version agility, as well as IDNA2008

• Variety of normalization options, that is, NFC, NFD, NFKC and NFKD

• More generic to strings
  • Two classes of strings are supported
  • IdentifierClass
  • FreeformClass (display name, password, etc.)
Who uses PRECIS?

• You 😊

• Especially, if your protocol is new to the IETF, and it defines identifiers and/or names, be sure that i18n is mandatory
How to use PRECIS?

• Figure out which of your protocol elements are user-facing
• Those are the only ones you should internationalize

• Figure out which of those elements are identifiers
  • If it is a string to designate a certain people / place / resource / service on the Internet, use IdentifierClass
  • Otherwise, use FreeformClass

• Define your PRECIS profile and register it to IANA

• Consult existing profiles, such as RFC 7613, 7700
Remaining Issues
Unicode, again

• Until Unicode 6.3.0, composition of combination mark has its own code point
• It was broken by Unicode 7.0.0

• ARABIC LETTER BEH WITH HAMZA ABOVE vs ARABIC LETTER BEH and ARABIC HAMZA ABOVE

• IETF is discussing this issue in LUCID list
  • https://www.ietf.org/mailman/listinfo/lucid
Variants

• Some languages / Scripts have variants
• Variants are the same pronunciation and meaning characters with different code point

• Tipically, Simplified Chinese vs Traditional Chinese
  • 咸 vs 亜

• Operational solution for variants in is discussed in LAGER WG
  • http://datatracker.ietf.org/wg/lager/charter/
Homographs

• Some scripts have similar looking characters each other
  • Pay vs Pay (ASCII vs Cyrillic)

• There is no solution in protocol (yet?)

• Practically, prohibiting mixed script string at registration is recommended
  • Consult RFC 6912 for more information
Summary
Summary

• i18n has to be designed in protocol

• PRECIS guides you

• i18n is still ongoing work

Any of your contribution to i18n work is highly appreciated
Your feedback please!

Please visit

https://www.surveymonkey.com/r/94precis