

# Precis Framework

draft-blanchet-precis-framework-00

Marc Blanchet  
Viagénie

[marc.blanchet@viagenie.ca](mailto:marc.blanchet@viagenie.ca)

# Rationale for the document

- Define a candidate replacement solution for stringprep
- Objectives of the solution:
  - combination of stringprep idea with idnabis inclusive algo
  - helping protocol designers for i18n (doing all the work for them)
  - helping move stringprep profiles using new solution
  - Minimize the need to create specific profiles by having a small buffet of choices that would satisfy most protocols.
  - target a single library (in OS) that would do all string preparation for all protocols.

# Approach

- Understand and characterize in detail what current stringprep profiles are using/doing.
- Then identify if some commonality is found and how it could be implemented using similar algorithm than IDNAbis.

# Comparing Stringprep Profiles

- (X.Y refers to RFC3454 annexes)
- Foreach profile:
  - Mapping: B.1, B.2 or special
  - Normalization: NFKC or NONE
  - Prohibited codepoints: C.1.1, C.1.2, C.2.1, C.2.2, C.3, C.4, C.5, C.6, C.7, C.8, C.9, or special
  - Additional prohibited codepoints from ASCII (punctuation for example)
  - Bidi processing: as section 6 or none
  - Unassigned U3.2 codepoints: as of A.1
- Have not considered nameprep since done with idnabis

# Mapping

- B.1 (Commonly mapped to nothing):
  - All except SASL trace and LDAP(special rules)
- B.2 (Mapping for case-folding used with NFKC):
  - iSCSI, XMPP nodeprep, LDAP
- C.1.2 (Non-ASCII space characters):
  - SASLprep (map to space)

# Prohibited Output

- C.1.1(ASCII space characters):
  - iSCSI and XMPP Nodeprep
- C.1.2(Non-ASCII space characters):
  - all except LDAP, Policy MIB, SASL trace
- C.2.1(ASCII control characters):
  - all except LDAP
- C.2.2(Non-ASCII control characters):
  - all except LDAP

# Prohibited Output

- C.3(Private use):
  - all
- C.4(Non-character code points):
  - all
- C.5(Surrogate codes):
  - all
- C.6(Inappropriate for plain text):
  - all except LDAP
-

# Prohibited Output

- C.7(Inappropriate for canonical representation):
  - all except LDAP, SASL trace
- C.8(Change display properties or are deprecated):
  - all
- C.9(Tagging characters):
  - all except LDAP

# Prohibited Output

- XMPP Nodeprep: " & ' / : < > @
- iSCSI: 21-2C, 2F, 3B-40, 5B-60, 7B-7F: ! " #  
\$ % & ' ( ) \* + , / ; < = > ? @ [ \  
] ^ \_ ` { | } ~
-

# Others

- Bidi:
  - Section 6 of Stringprep: all except Policy MIB and LDAP
- Unassigned code points:
  - A.1: all
- Normalization:
  - NFKC: all except SASL trace(none)

# Consolidation

- Forgetting for a moment LDAP and SASL trace which are more different than the others.
-

# Mapping

- Mapping:
  - B.1 (Commonly mapped to nothing): All
  - B.2 (Mapping for case-folding used with NFKC):  
iSCSI, XMPP nodeprep

# Prohibited Output

- C.1.1(ASCII space characters): iSCSI and XMPP Nodeprep
- C.1.2(Non-ASCII space characters): all except Policy MIB
- C.2.1(ASCII control characters): all
- C.2.2(Non-ASCII control characters): all
- C.3(Private use): all
- C.4(Non-character code points): all
- C.5(Surrogate codes): all
- C.6(Inappropriate for plain text): all
- C.7(Inappropriate for canonical representation): all
- C.8(Change display properties or are deprecated): all
- C.9(Tagging characters): all
- XMPP Nodeprep: " & ' / : < > @
- iSCSI: 21-2C, 2F, 3B-40, 5B-60, 7B-7F: ! " # \$ % & ' ( ) \* + , / ; : :13  
< = > ? @ [ \ ] ^ \_ ` { | } ~

# Others

- Bidi:
  - Section 6 of Stringprep: all except Policy MIB
- Unassigned code points:
  - A.1: all
- Normalization:
  - NFKC: all

# Differentiators

- Roughly:
  - Space
  - case-folding
  - Some non-letter ascii chars not allowed

# Grouping

- Common to all: B.1, C.1.2, C.2.1, C.2.2, C.3, C.4, C.5, C.6, C.7, C.8, C.9, NFKC
- xmpp nodeprep and iscsi are identical except some ascii punctuation
  - => case mapping, no space, restricted ascii
  - Defined as: restricted internationalized identifier (RiID)
- xmpp resourceprep and policy mib are identical except non-ascii space
  - => no case mapping, space, almost any printable ascii
  - Defined as: less restricted internationalized identifier (LRiID)

# Proposal

- Define two classes of internationalized strings, as per the grouping above
  - restricted internationalized identifier (RiID)
  - less restricted internationalized identifier (LRiID)
- Satisfies 4 of the 6 current profiles.
- Hopefully be picked by other protocols.
- Towards the objectives cited initially
-

# Implementation Summary

- Modify the IDNAbis tables algorithm:
  - Define RiID\_VALID, LRiID\_VALID
  - Add new rules for these new
  - For RiIDs: RiID\_VALID and P\_VALID are valid
  - For LRiIDs: LriID\_VALID and P\_VALID are valid
  - Same for RiID\_DISALLOWED, LRiID\_DISALLOWED
  - Add new generic rule for prohibiting additional codepoints specific to protocols. (the codepoints being defined in the « client » protocol spec)
  - Add case mapping for RiID
  - Specify NFKC as required

# Additional Considerations

- Bidi separator: if the identifier has a separator where Bidi has to be done separately on each part, then it is defined in the profile as « BIDI separator ».
- (credits to Yoshiro Yoneya who brought that issue)

# Protocols using Precis Framework

- The « customer » protocol of the precis framework would then:
  - Choose a string class: RiID, LRiID
  - Lists if any additional codepoints are prohibited (ex: XMPP and iSCSI ASCII « punctuation » codepoints)
  - Lists the Bidi separator(s)
- (hopefully, these choices could be implemented are arguments to a string\_prepare function/method in the OS)

# SASL trace and LDAP

- Forgot « for a moment »
- would require specific profiles as before.

# Backward Compatibility

- Author is currently writing a parser of various tables to create the diffs between the tables of each stringprep profile and this proposal.
- Hopefully not too bad. Most likely similar to known incompatibilities of IDNAbis.
- More later.

# Conclusion and Next Steps

- Aim to describe a proposal for a replacement of stringprep based on a list of objectives
- Proposed next steps in no specific order:
  - Find a co-editor
  - Detail the new rules
  - Finish writing parsers to compare stringprep profiles and this proposal codepoints tables for Unicode 5.2. Would make clear the backward compatibility
  - Adopt as Precis Working Group document

# Questions?

draft-blanchet-precis-framework-00

Contact info: [marc.blanchet@viagenie.ca](mailto:marc.blanchet@viagenie.ca)