CLDR - Person Name Validation

Michael McKenna - UTW October 2024

2024-10-22

Problem

CLDR Person Name

→ Name formatting

→ But no name validation

given: Jane
surname: Doe

given: 駿
surname: 宮崎

vs.

given: 𝔍𝗮ne
surname: Dddóöoôòe

given: !駿<<
surname: 宮🐲崎

Solution - validation

→ Simple validation

→ Different levels for different scenarios

→ Customizable to adapt to specific needs

→ Close to “normal” expectations by language

Solution - getting there

→ Name data analysis

→ Existing software application practices

→ Research on name characteristics

→ Modern naming practices

→ Regulatory constraints

→ CLDR contributor feedback

Validation levels

→ Strict

→ Customized

→ Lenient

→ Minimal

Strict Validation

- Expectations for names in a locale
- People move and keep names
- Validate for “affinity groups” of locales
  - TC: most major languages
  - DDL: Digitally Disadvantaged Languages
- Examples
  - TC Latin
  - Han | Japanese
  - Cyrillic | Greek | Arabic | Hebrew
  - Thai | Devanagari | Tamil | ...
- DDL
  - Union of TC + DDL language validation attributes

Tailored Validation

- → Customized by client
  - ⟾ Specific set of characters
  - ⟾ Specific set of punctuation
  - ⟾ Possible length requirements
- → Example scenarios
  - ⟾ Passports
  - ⟾ Government registries
    - ⇢ e.g. UK Deed Poll (allowed characters, punctuation, length)
    - ⇢ e.g. US naming laws (no diacritics in California)
    - ⇢ e.g. JP name registration (Jimmeiyo Kanji, katakana, punctuation)
  - ⟾ Compliance onboarding
    - ⇢ specific to contracted service requirements

Lenient Validation

Used when more flexibility is required

- → May need more than one script in entry field
  - ⟾ e.g.

    given: “Hayao (駿)”
    surname: “Miyazaki (宮崎)”
- → Allows
  - ⟾ All *letters* for all scripts
  - ⟾ Most punctuation - note: can allow “unsafe” sequences such as <script>
  - ⟾ No emoji
- → But want to avoid confusables and obfuscation
  - ⟾ example rules
    - ⇢ Cannot mix scripts in one string of letters or “token” - “Δoe”
    - ⇢ Must have letters `\p{Letter}` in name fields
    - ⇢ Cannot be all punctuation

Minimal Validation

Bare bones validation.

→ Minimal validation
→ Valid Unicode
→ No control or formatting characters
→ Must have at least one letter in each populated field
→ “Invisible” ZWSP and some RTL direction controls allowed in context with certain scripts

given: “Jane123🥳”

surname: “Doea҉!”

Assumptions

Name data to be validated:

→ All html/xml entities, NCRs resolved

→ Normalized to NFC

→ Data security to be done by the client

⟾ UTR #36 Unicode Security Considerations

⟾ OWASP (Open Worldwide Application Security Project)

Normalization Forms

- NFC - Composed form. Primary form used in analysis
- NFD - Decomposed form. Used is specific validation steps

Name Fields

	Strict	Customized	Lenient	Minimal
Not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓

At least one primary name field must be populated

given or **surname**

If validating a stand-alone single string, it must not be empty or null.

Client needs to set requirements if it needs specific name field to be required.

Valid Unicode, not invisible

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓

All data must be valid, defined Unicode

No control characters \p{Cc}

No “invisible” characters

specials, language tags, most format characters

Special cases allowed, by script

E.g. ideographic variation selectors, RTL

Analyze data by “Tokens”

A name field is broken into “tokens”

UAX #29 - Unicode Text Segmentation

→ Groups of letters and non-letters

Example: “Miyazaki (宮崎)” ⇒ “Miyazaki| |(|宮崎|)| ”

2 letter group tokens, 3 non-letter tokens

Example: “van O’Leary” ⇒ “van| |O’Leary| ”

2 letter tokens, 1 non-letter (apostrophe recognized)

Names have letters

At least one token in each field must be letters

“^\p{Letter}+$”

OK: “Jane Doe” | “جین ڈو” | “ジェーン・ドゥ”

Not OK: “;;, ,;;” | “🤣①②③” | “▪ = °—”

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓

Non-spacing letters

Non-spacing letters must be appropriate for surrounding script

- CJK variation selectors
- RTL script → formatters

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓

Repeating Non-Letters

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓
repeating non-letters	✓	✓	✓	

Except for opening and closing punctuation “(([[{{}}]]))”, you should (usually) not see repeating non-letters.

\p{P}-\p{Po}\p{Pe}

Language data:

→ *maxIdenticalNonLetters* { count }
→ *maxRepeatingNonLetters* { count }
→ *allowedRepeatingNonLetters* { set }

(not checked for *minimal* validation)

Punctuation

→ Non-letter punctuation at beginning or end of field

→ Balanced start \p{Ps} and end \p{Pe} punctuation

OK: “Miyazaki (宮崎)” | “Mhd.” | “Thos., Esq.”

Not OK: “,Miyazaki)宮崎(“

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓
repeating non-letters	✓	✓	✓	
punctuation placement	✓	✓	✓	
balanced punctuation	✓	✓	✓	

Combining Marks

Normalize to NFD

→ No combining mark by itself

→ No combining mark after non-letters

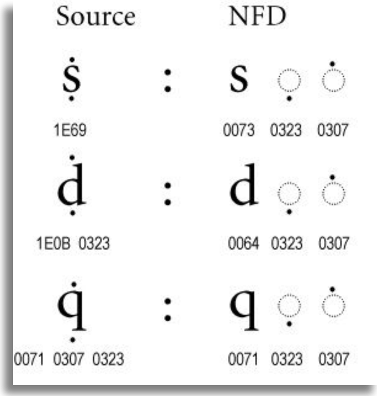
Not OK: “◌́ abc” | O’◌́ Leary”.

→ No repeating identical combining marks

⟾ e.g. two acute accents in a row ◌́ ◌́

maxContiguousCombiningMarks

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓
repeating non-letters	✓	✓	✓	
punctuation placement	✓	✓	✓	
balanced punctuation	✓	✓	✓	
repeating combining marks	✓	✓	✓	



Source		NFD
ṩ 1E69	:	s ◌̣ ◌̇ 0073 0323 0307
ḍ̇ 1E0B 0323	:	d ◌̣ ◌̇ 0064 0323 0307
q̣̇ 0071 0307 0323	:	q ◌̣ ◌̇ 0071 0323 0307

Combining Marks

Abugida script-languages have many combining marks

(Mymr, Tibet, Deva, ...)

Devanagiri: छोड़ों

091B DEVANAGARI LETTER CHA

094B DEVANAGARI VOWEL SIGN O

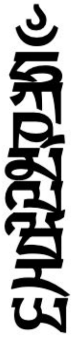
0921 DEVANAGARI LETTER DDA

093C DEVANAGARI SIGN NUKTA

094B DEVANAGARI VOWEL SIGN O

0902 DEVANAGARI SIGN ANUSVARA

Tibetan: HAKṢHMALAWARAYAṀ



ཧ 0F67 TIBETAN LETTER HA

◌ྐ 0F90 TIBETAN SUBJOINED LETTER KA

◌ྵ 0FB5 TIBETAN SUBJOINED LETTER SSA

◌ྨ 0FA8 TIBETAN SUBJOINED LETTER MA

◌ླ 0FB3 TIBETAN SUBJOINED LETTER LA

◌ྺ 0FBA TIBETAN SUBJOINED LETTER FIXED-FORM WA

◌ྼ 0FBC TIBETAN SUBJOINED LETTER FIXED-FORM RA

◌ྻ 0FBB TIBETAN SUBJOINED LETTER FIXED-FORM YA

◌ྂ 0F82 TIBETAN SIGN NYI ZLA NAA DA

Repeating Letters

Strip out accents after NFD

“Jàáaâäñne” → “Jaaaaanne”

maxRepeatingBaseLetters { count }

allowedRepeatingBaseLetters { array }

OK: “Grossschadel”

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓
repeating non-letters	✓	✓	✓	
punctuation placement	✓	✓	✓	
balanced punctuation	✓	✓	✓	
repeating combining marks	✓	✓	✓	
repeating letters	✓	✓	✓	

One script per token

Letter tokens must be in one script

OK: “Miyazaki (宮崎)”

not OK: “GalגלGadotגדות” (one token)

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓
repeating non-letters	✓	✓	✓	
punctuation placement	✓	✓	✓	
balanced punctuation	✓	✓	✓	
repeating combining marks	✓	✓	✓	
repeating letters	✓	✓	✓	
one script per token			✓	

Exemplar Characters

CLDR Exemplar sets of characters used for writing a language

→ Letters

en: abcdefghijklmnopqrstuvwxyz àáâãäåæç
èéêëìíîïñòóô öøùúûüÿāă ēĕīĭ ōŏœ ūŭ

fr: abcdefghijklmnopqrstuvwxyzßàáâãäåæç
èéêëìíîïñòóôõöøùúûüÿā ćē ī ĳ œřšſ ŭ

→ Punctuation

⟾ e.g.: en: [! " # & ' ( ) * , - . / : ; ? @ [ ] § ‐ ‑ – — ‘ ’ “ ” † ‡ … ′ ″]

⟾ e.g.: fr: [! " # & ( ) * , - . / : ; ? @ [ ] § « » ‐ ‑ – — ’ “ ” † ‡ …]

Exemplar affinity sets

Union of exemplar letters and punctuation

→ related languages
→ use same script

TC: All Latn “modern” languages:
\p{exemplar=/^(az|ca|cs|da|de|es|fi|fil|fr|hr|hu|is|it|lt|lv|ms|nl|no|pt|ro|sk|sl|sv|tr|pl|cy|ga|bs|et|tk|id|vi|uz|en|gl|sw|af|eu|nn|so|jv|sq|zu)$/}

“[a-z·ß-öø-ÿāăąćčďđēėęěğģĩīį-ıķĺļľłńņňőœŕřśşšťũūůűųŵŷźżžơưșțə‘’ẁẃẅạảấầẩẫậắằẳẵặẹẻẽếềểễệỉịọỏốồổỗộớờởỡợụủứừửữựỳỵỷỹ]”

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓
repeating non-letters	✓	✓	✓	
punctuation placement	✓	✓	✓	
balanced punctuation	✓	✓	✓	
repeating combining marks	✓	✓	✓	
repeating letters	✓	✓	✓	
one script per token			✓	
character Affinity Sets	✓			

All Cyrl:
“[’а-яё-ќўџґғқңүұһәө]”

Exemplar tailored sets

en-Latn-US-x-California:

“[a-z]”

en-Latn-GB-x-Deedpoll:

“[a-z ’ à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ø ù ú û ü ý þ ÿ ā ă ą ć ĉ ċ č ď đ ē ĕ ė ę ě ĝ ğ ġ ģ ĥ ħ ĩ ī ĭ į ı ĳ ĵ ķ ĸ ĺ ļ ľ ŀ ł ń ņ ň ŉ ŋ ō ŏ ő œ ŕ ŗ ř ś ŝ ş š ß ſ ţ ť ŧ ũ ū ŭ ů ű ų ŵ ŷ ź ż ž]”

	Strict	Customized	Lenient	Minimal
not empty	✓	✓	✓	✓
given or surname	✓	✓	✓	✓
valid Unicode	✓	✓	✓	✓
control characters	✓	✓	✓	✓
invisible characters	✓	✓	✓	✓
one token must be letters	✓	✓	✓	✓
non-spacing characters	✓	✓	✓	✓
repeating non-letters	✓	✓	✓	
punctuation placement	✓	✓	✓	
balanced punctuation	✓	✓	✓	
repeating combining marks	✓	✓	✓	
repeating letters	✓	✓	✓	
one script per token			✓	
character Affinity Sets	✓			
customizable Sets		✓		

What we are *not* doing

- No vowels check?
  - “Twm”, “Tylhr”, “Ng”
  - ‘y’, ‘h’, ‘w’, ‘r’, ‘j’ act as vowels in some language
- Single letter check?
  - “Malcom X”
  - “Harry S Truman”
  - “Karen O”
- Offensive terms
  - Language dependent
  - May be real names in some locales
- Statistical name pattern / gibberish checks?
  - Requires in-depth analysis
  - Large data requirement or remote service

Next steps

- Complete proposal
- Data structure definition
- Prototype
- Review, feedback

Thank you

【
U+3010

シ
U+30B7

メ
U+30E1

☎
U+260E

デ
U+30C7

👽
U+1F47D

☂
U+2602

个
U+4E2A

😌
U+1F60C

Ψ
U+03A8

غ
U+063A

ミ
U+30DF

ε
U+03B5

〇
U+3007

♺
U+267A

ζ
U+03B6

く
U+304F

ദ
U+0D26

UNICODE

Thank you to our Organizational Members

Microsoft

Adobe

Meta

NETFLIX

salesforce

airbnb

Bloomberg

translated.

Google

amazon

Stanford
SILICON
Advancing Digitally
Disadvantaged Languages

sas

WIKIMEDIA
FOUNDATION

ORACLE

Spotify

Berkeley
University of California

IBM

Bretagne
Numérique
.BZH FONDS DE DOTATION

SIL

DECOtype

Yat

workday.

中 易
BEIJING ZHONGYI
ELECTRONICS CO.

Emojipedia

إبانة
ebanah

文林
Wénlín

BASIS
TECHNOLOGY

Thank you to our Industry and Media Partners

GALA
Globalization & Localization Association

MultiLingual

WOMEN IN
LOCALIZATION