



Internationalized Domain Names

How It Works

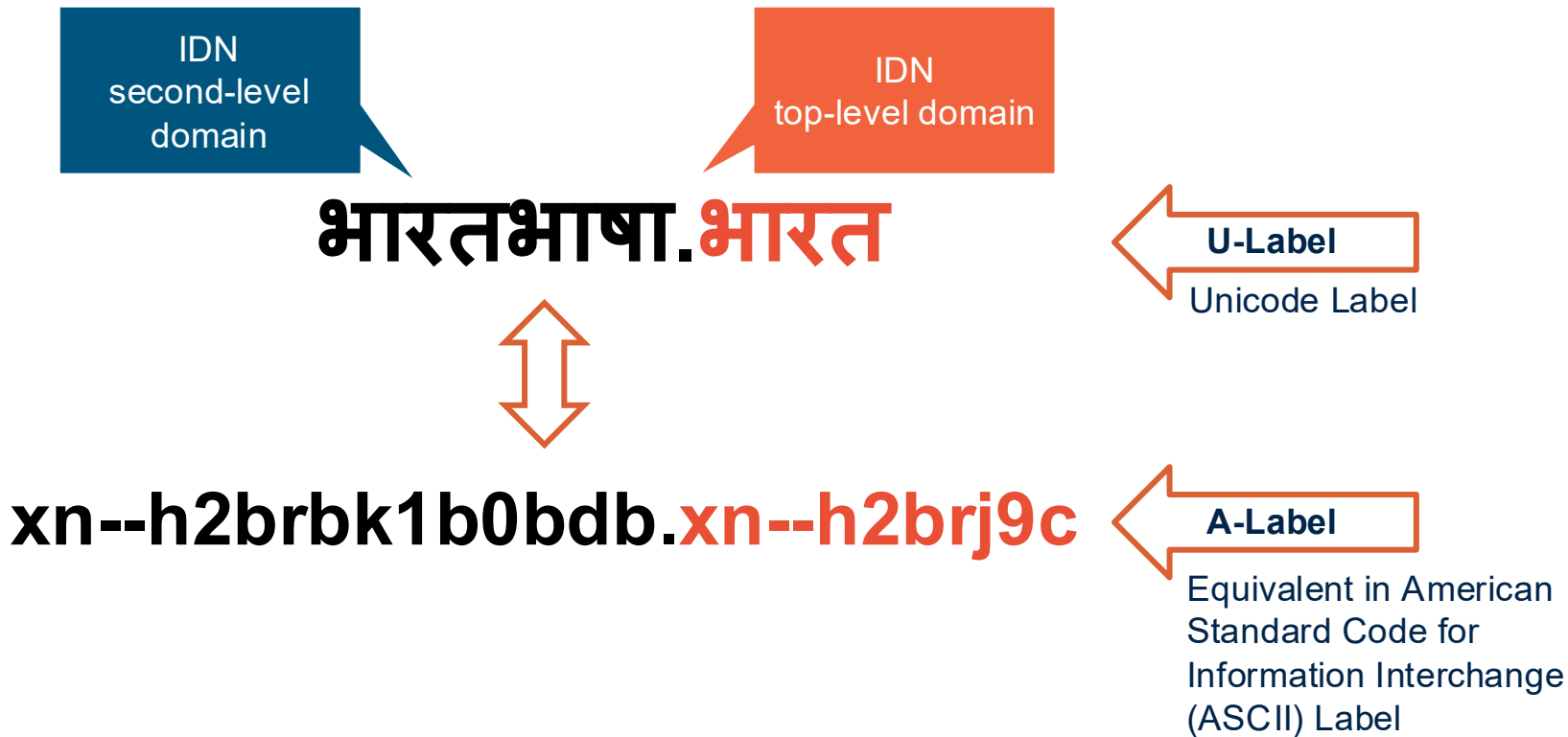
10 March 2026

Agenda

- The process to enable IDNs in local languages and scripts
- Documenting the rules for IDNs
- Reference LGRs for First Nation languages in Canada
- Reference LGRs for Indonesian scripts
- Q & A

The Process to Enable IDNs in Local Languages and Scripts

Internationalized Domain Name (IDN) Labels



IDNs for Applications (IDNA) 2008

- Internationalized Domain Names for Applications (IDNA) is a set of standards for handling Internationalized Domain Names (IDNs). Developed by Internet Engineering Task Force (IETF).
- The current version is IDNA 2008, consisting of:
 - Standard Track:***
 - RFC 5890 IDNA: Definitions and Document Framework
 - RFC 5891 IDNA: Protocol
 - RFC 5892 The Unicode Code Points and IDNA
 - RFC 5893 Right-to-Left Scripts for IDNA
 - Informational Track:***
 - RFC 5894 Background, Explanation, and Rationale
 - RFC 5895 Mapping Characters for IDNs in Applications (IDNA) 2008
- All IDNs must be IDNA 2008 compliant.

RFC 5892: The Unicode Code Points and IDNA

The value of the property (of a Unicode Code Point) is to be interpreted as follows.

PROTOCOL VALID (PVALID)*

Those that are
allowed to be used in
IDNs.

Example:

“é”

U+00E9

CONTEXTUAL RULE REQUIRED

Not to be used in labels
unless specific context
is met.

Example:

“l-l”

U+006C U+00B7 U+006C

(U+00B7 must be
between 'l' (U+006C))

DISALLOWED

Those that should
clearly not be included
in IDNs.

Example:

“?”

U+003F
(Question Mark)

UNASSIGNED

Code points that are
not designated in the
Unicode Standard.

Example:

U+05C8 -
U+05CF

*A label consists only of PVALID code points does not imply that the label can be used in DNS.

Additional Requirements in IDNA2008

- **RFC 5892, Introduction section states:**
“...However, that a label consists only of code points that have this property value [PVALID] does not imply that the label can be used in DNS.”
- **RFC 5890, Section 2.3.2.3 states:**
*“Because of the diversity of characters that can be used in a U-label and the confusion they might cause, such restrictions [**“variant definitions and rules beyond those imposed by DNS or IDNA”**] are mandatory [emphasis added] for IDN registries and zones even though the particular restrictions are not part of these specifications..”*
- **Additional restrictions beyond IDNA are mandatory, therefore, the Label Generation Rules (LGRs)**

Label Generation Rules (LGRs)

Label Generation Rule (LGR)



Metadata

Version, Date, Language, Description, Reference, etc.



Code Point Repertoire

Code Points, Code Point Sequences



Variants

Variant Code Points and Their Types



Rules

Contextual Rules, Whole Label Rules, Action Rules

Designing Principles

Starting from [RFC 6912](#) - Principles for Unicode Code Point Inclusion in Labels in the DNS:

- **Longevity** – stable across Unicode versions
- **Least Astonishment** – take into account the population using a code point
- **Contextual Safety** – sensitive to ways in which code point may be used in malicious ways
- **Conservatism** – any code point inclusion decision is as conservative as practicable
- **Inclusion** – default is excluded, then add code point which is safe based on usability and confusability
- **Simplicity** – rules determining use should be simple to understand
- **Predictability** – rules determining whether a code point is included are predictable for others to reach the same conclusion
- **Stability** – if permitted, taking it out very hard



Metadata



Code Point
Repertoire



Variants



Rules

Code Point Repertoire

What are code points to be used?



Metadata



Code Point
Repertoire



Variants



Rules

Code Point Repertoire - What is the Goal?

- **To create a mnemonic system for use in the Domain Name System (DNS)**
 - A mechanism to remember IP address
 - Must remain secure and stable in use – if DNS is confusing to users, then the motivation is not met
- **Not required to completely cover a language or a script**
 - May not formed a word in a language
 - Not restricted to “correct” spellings
 - May not carry a meaning in the “lexical” sense



Metadata



Code Point
Repertoire



Variants



Rules

ASCII Domain Name Label

www.cafe-123.com

Forming ASCII Labels at Second-Level, Third Level

Use **LDH**

- Letters [a - z]
- Digits [0 - 9]
- Hyphen [H]

Label length = 63

Other constraints e.g. on Hyphen

Forming ASCII Label at Top-Level Domain

Use **only Letters**

- Letters [a - z]

Label length = 63



Metadata



Code Point
Repertoire



Variants



Rules

Domain Name Mnemonics in ASCII

- Top-Level Domain
 - Letters [a - z]
- Second-Level use LDH
 - Letters [a - z]
 - Digits [0 - 9]
 - Hyphen (H)

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON		1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del



Metadata



Code Point
Repertoire



Variants



Rules

Repertoire for IDN Mnemonics

- Unicode Standards includes code charts for all encoded scripts.

	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط
1	ق	ك	ل	م	ن	ه	و	ز	ح	خ	د	ذ	ر	ز	س	ش
2	ص	ض	ط	ق	ك	ل	م	ن	ه	و	ز	ح	خ	د	ذ	ر
3	ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط
4	ق	ك	ل	م	ن	ه	و	ز	ح	خ	د	ذ	ر	ز	س	ش
5	ص	ض	ط	ق	ك	ل	م	ن	ه	و	ز	ح	خ	د	ذ	ر
6	ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط
7	ق	ك	ل	م	ن	ه	و	ز	ح	خ	د	ذ	ر	ز	س	ش

Partial Arabic script Unicode code chart

	120	121	122	123	124	125	126	127	128	129	12A	12B
0	ሀ	ሐ	ሠ	ሰ	ቀ	ቐ	በ	ተ	ኀ	ነ	አ	ከ
1	ሁ	ሑ	ሡ	ሱ	ቂ	ቐ	ቡ	ቱ	ኁ	ኑ	ኦ	
2	ሂ	ሐ	ሢ	ሲ	ቃ	ቐ	ቢ	ቲ	ኂ	ኑ	ኦ	ከ
3	ሃ	ሐ	ሣ	ሳ	ቄ	ቐ	ባ	ታ	ኃ	ና	አ	ከ
4	ሄ	ሐ	ሤ	ሴ	ቅ	ቐ	ቤ	ቲ	ኄ	ኑ	ኦ	ከ
5	ህ	ሐ	ሥ	ሴ	ቆ	ቐ	ብ	ቲ	ኅ	ን	እ	ከ
6	ሆ	ሐ	ሦ	ሶ	ቇ	ቐ	ቦ	ቲ	ኆ	ኑ	ኦ	
7	ሰ	ሐ	ሸ	ሴ	ቈ		ቦ	ቲ	ኇ	ና	እ	

Partial Ethiopic script Unicode code chart



Metadata



Code Point
Repertoire

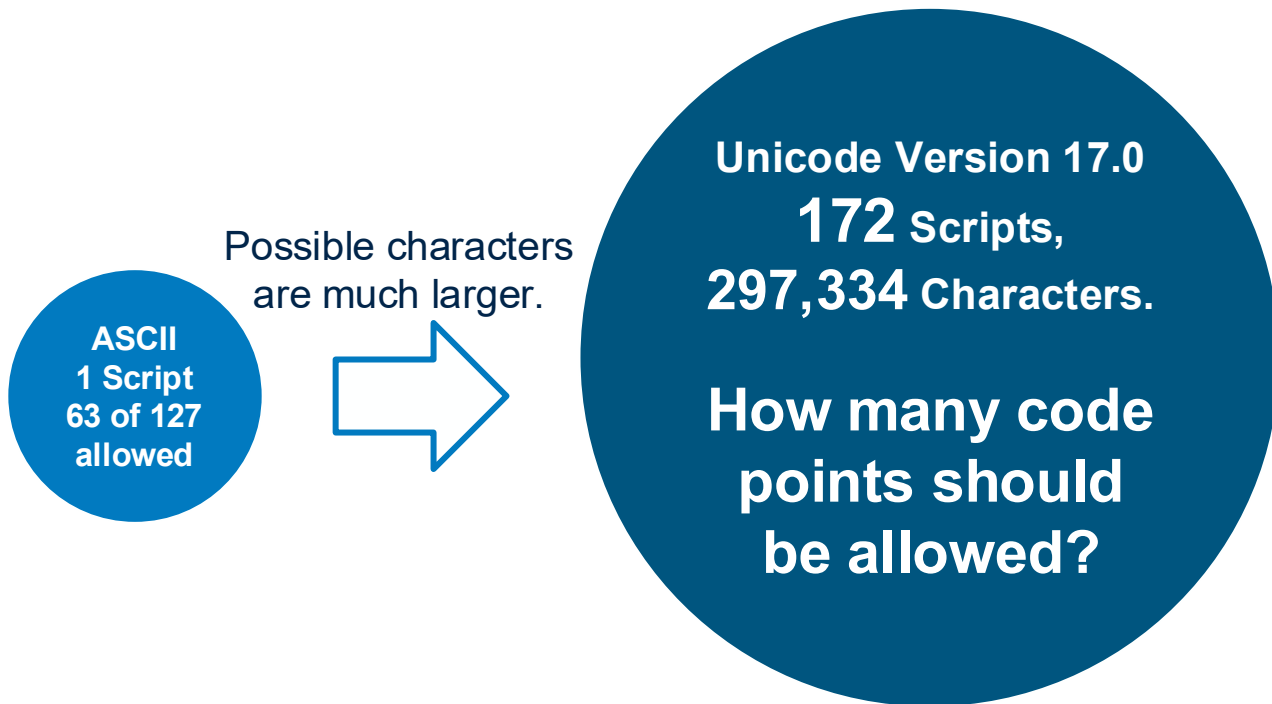


Variants



Rules

ASCII vs Unicode



Metadata



Code Point
Repertoire



Variants



Rules

Building the Code Point Repertoire

Step 1: Review relevant script Unicode code chart(s)

Step 2: Before including a code point, ask the following questions:

- Is it PVALID?
- Is it used with the language or script in scope?
- Is it suitable in identifiers?
 - Is it in widespread modern use?
 - Is it not technical / religious / limited use only?
 - Is it not really a punctuation / symbol?
 - Is it really necessary for representing identifiers?
- Is the Unicode encoding of the code point stable?
 - Are there any rendering issues?
- How accessible would a TLD containing that code point be?
 - Are there input/keyboard concerns?
- Is it in tension with any of the RFC6912 Principles in any way?



Metadata



Code Point
Repertoire



Variants



Rules

***Are there any code points considered “same”
by the language or script community?***



Metadata



Code Point
Repertoire



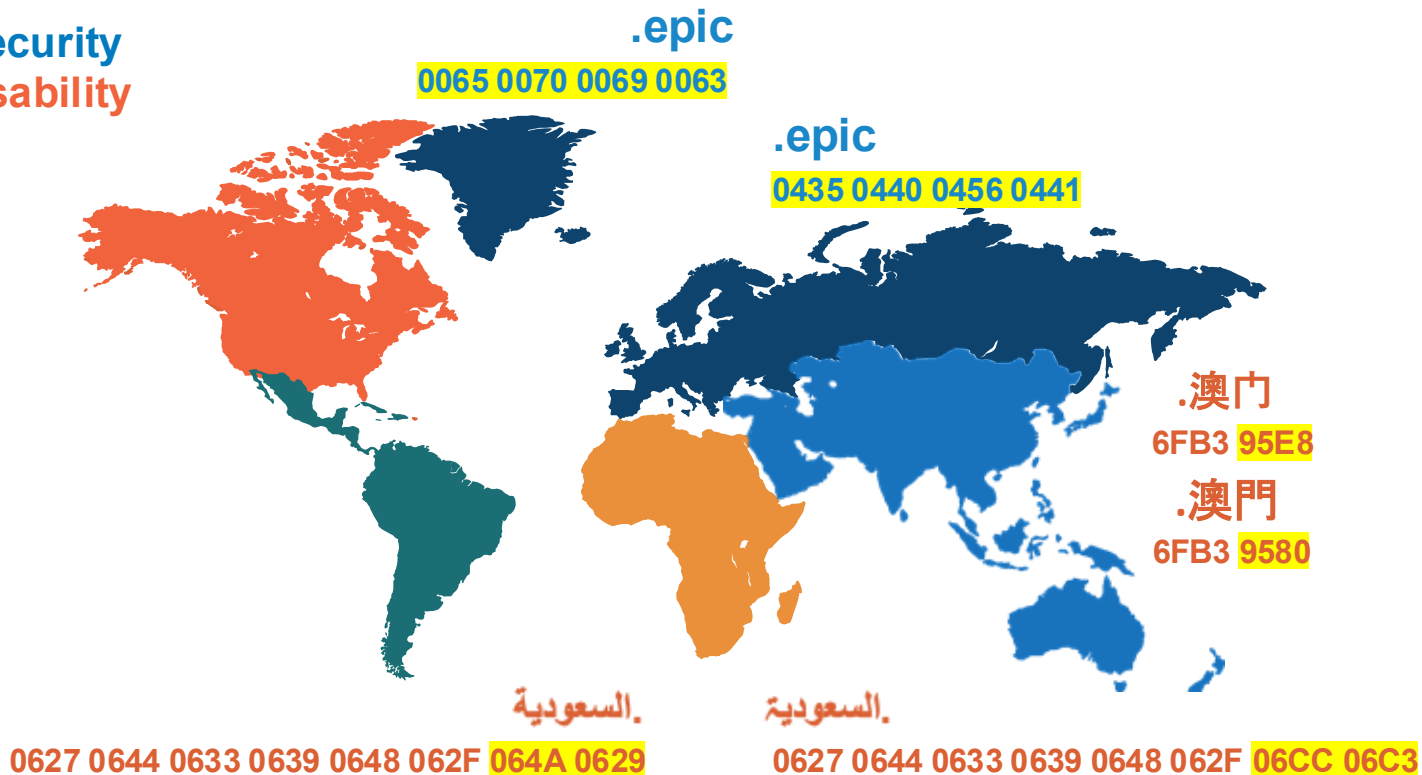
Variants



Rules

Understanding Variants

- Security
- Usability



Metadata



Code Point
Repertoire



Variants



Rules

Variants - What is the Goal?

- **Make the mnemonic system to minimize user confusion**
 - Defining code point or code point sequences causing two (or more) labels functionally “the same” within script, and cross-script.
- **Conservatism requires:**
 - maximize “blocked” variants
 - minimize “allocatable” variants
- **Note: Successfully defining variant rules for an LGR is not trivial**



Metadata



Code Point
Repertoire



Variants



Rules

Defining Variants - Questions to Ask

- Would a reasonable person with native knowledge of the script consider a pair of code points interchangeable?
- Would such a person be unable to determine which of these interchangeable code points was used by appearance?
- Is there an alternative representation?
- Should any of the variants of this code point be contingent on context?
- Are any variants contemplated that are in tension with any of the Principles?
- Any cross-script variants?



Metadata



Code Point
Repertoire



Variants



Rules

Are there any restrictions required to ensure the language or script is used in a secure and stable manner in the DNS?



Metadata



Code Point
Repertoire



Variants



Rules

Rules - What is the Goal?

- **Reducing label space, preventing labels which should not be possible for various reasons**
 - Not licensed by the script (but not spelling rules)
 - Cause security issues
 - Cause usability constraints
 - Other?
- **Reducing allocatable label by making them blocked in certain cases**
- **Put in contextual contexts for code points or their sequences**



Metadata



Code Point
Repertoire



Variants



Rules

Example of Risk Mitigation (Thai Script)



Metadata



Code Point
Repertoire



Variants



Rules

- Double U+0E40 (Thai Character Sara E) look like a U+0E41 (Thai Character Sara Ae)
- Thai script LGR includes a rule to prevent U+0E40 to follow another U+0E40

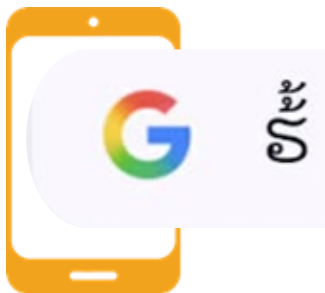
Example of Risk Mitigation (Laos Script)

“U+0EAE (ສ) U+0EC9(ັ) U+0EC9(ັ)”

Consonant

Tone Mark

Tone Mark



Chrome on Android 16



Chrome on iOS 26.3

- The Laos script expects none or one tone mark on a consonant.
- More than one tone mark on a consonant are rendered inconsistently.
- Laos script LGR includes a rule to restrict one tone mark on a consonant.



Metadata



Code Point
Repertoire



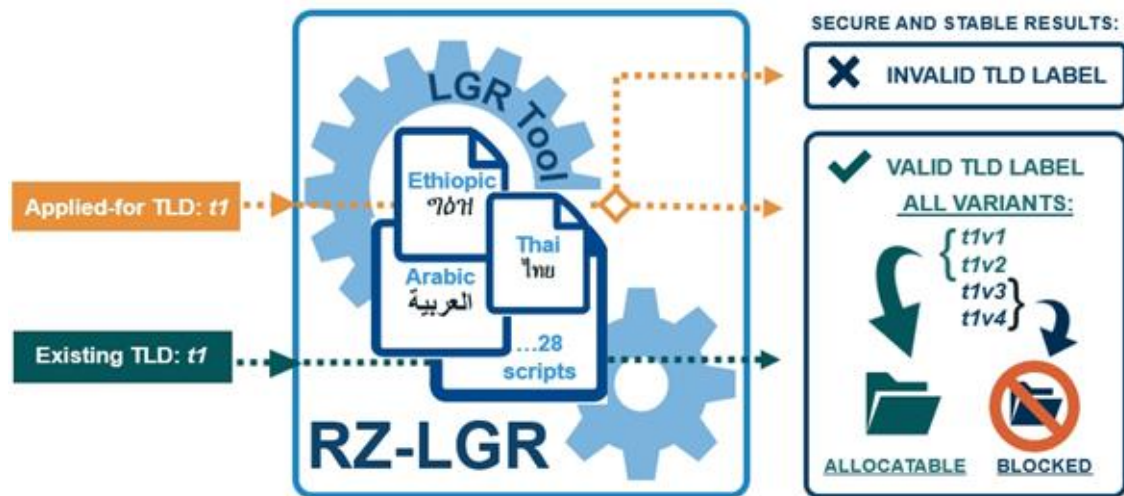
Variants



Rules

Usage of LGR

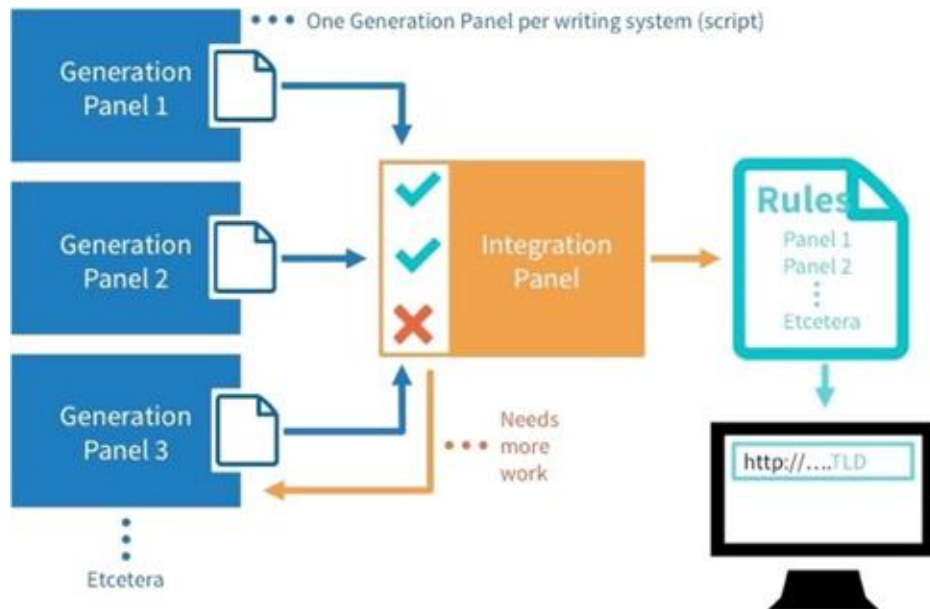
- LGR in machine-readable format can be processed by the LGR Tool to validate a label and generate or validate their variant labels
- For example: Root Zone Label Generation Rules



Need for Community-Driven Development of Rules for IDNs

Root Zone Label Generation Rules (RZ-LGR)

- RZ-LGR development process includes input from script communities
- RZ-LGR version 6
 - 27 scripts
 - supporting 387+ languages
- 11 years of work, 18 Generation Panels (GPs) from 45 countries
- 270+ script community volunteers
10,000+ volunteer hours



- RZ-LGR version 6 (RZ-LGR-6) includes 27 scripts
 - Arabic, Armenian, Bangla, Chinese (Han), Cyrillic, Devanagari, Ethiopic, Georgian, Greek, Gujarati, Gurmukhi, Hebrew, Japanese (Hiragana, Katakana, and Kanji [Han]), Kannada, Khmer, Korean (Hangul and Hanja [Han]), Lao, Latin, Malayalam, Myanmar, Oriya, Sinhala, Tamil, Telugu, Thaana, and Thai

Reference LGRs for Second-Level (Ref. LGR)

- ICANN org developed second level [reference LGRs](#) and available [RZ-LGR](#) following consultations with the script communities, which were finalized after Public Comment proceedings.
- TLD registry operators can use reference LGRs when they develop rules for IDN registration under their TLDs.
 - ICANN org uses reference LGRs to review the registration rules (called IDN Table) under gTLDs.
 - This contributes to consistency and transparency of the review as requested by the Registries Stakeholder Group (RySG).
- Reference LGR can be used develop an IDN Table which is required for submitting an IDN ccTLD Request via the IDN ccTLD Fast Track Process.

Reference LGRs for Second-Level (Ref. LGR)

ICANN publishes reference LGRs for 60 languages and scripts:

- **32 language-based reference LGRs:** Arabic, Belarusian, Bosnian (Cyrillic), Bosnian (Latin), Bulgarian, Chinese, Danish, English, Finnish, French, German, Hebrew, Hindi, Hungarian, Icelandic, Inuktitut, Italian, Japanese, Korean, Latvian, Lithuanian, Macedonian, Montenegrin, Norwegian, Polish, Portuguese, Russian, Serbian, Spanish, Swedish, Thai, and Ukrainian.
- **28 script-based reference LGRs:** Arabic, Armenian, Balinese, Bangla (Bengali), Chinese, Cyrillic, Devanagari, Ethiopic, Georgian, Greek, Gujarati, Gurmukhi, Hebrew, Japanese (Hiragana, Katakana, Kanji [Han]), Kannada, Khmer, Korean (Hangul, Hanja [Han]), Lao, Latin, Malayalam, Myanmar, Oriya, Sinhala, Tamil, Telugu, Thaana, and Thai.

Collaboration on Developing the Ref. LGR

Voluntary expertise provided by a community working group

- Linguistic expertise to develop a proposal which define repertoire, variants, rules.
- Word list or corpus for a language or script.
- Review and test the LGR.

Support provided by ICANN org

- Technical support on developing an LGR in RFC7940 format, based on input from the community.
- Taking an LGR through the public comment process before finalizing and publish an LGR.



How to Get Involved

- If you'd like to develop a reference LGR for your language or script, please contact IDNProgram@icann.org
- Information on IDNs, please visit icann.org/idn

Documenting the Rules for IDNs

Akshat Joshi
Neo-Brahmi Script Generation Panel Member

RFC 7940: Representing LGR Using XML



Metadata

Version, Date, Language, Description, Reference, etc.



Code Point Repertoire

Code Points, Code Point Sequences



Variants

Variant Code Points and Their Types



Rules

Contextual Rules, Whole Label Rules, Action Rules

- ICANN org published LGRs in both XML and equivalent HTML format
- ICANN org provide LGR Tool to build an LGR and generate XML and HTML

LGR Tool

- The LGR Tool, provided by ICANN org, helps with developing and managing an LGR, validating labels against a LGR, and reviewing IDN tables.
 - To use the tool, please visit: [***lgrtool.icann.org***](https://lgrtool.icann.org).
It is available for all users with an ICANN account.
 - User Guides are available on the ***LGR Tool*** [***webpage***](#).
- The LGR Tool is an open source application. Parties interested in integrating this functionality into their own systems can find the source code on github: [lgr-core](#), [lgr-django](#), [munidata](#), [picu](#)

Exercise: Building a “A-G” Language LGR

Given that the “A-G” language is a mock-up language with following characteristics:

1. The “A-G” language contains following 11 Latin script letters and 1 sign: a, á, b, c, d, e, é, f, *f*, g, ‡(U+01C2), and ° (U+00B0).
2. The two forms of ‘f’ (f, *f*) are considered “same” by the language community. For example, the label “fin” can be confused with “*fin*”. If both versions are registered to different registrants, it will cause security issues.
3. The ‡ must not be at the beginning of any label.



Metadata



Code Point
Repertoire



Variants



Rules

Exercise: “A-G” Language Repertoire

1. The “A-G” language contains following 11 Latin script letters and 1 sign:
a, á, b, c, d, e, é, f, g, †(U+01C2), and ° (U+00B0).

Elevent (11) Code points to be included, ° (U+00B0) is excluded.

	000	001	002	003	004	005	006	007
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x

Basic Latin

	008	009	00A	00B	00C	00D	00E	00F
0	XXX	DC5	NB SP	°	À	Đ	à	đ
1	XXX	PU1	ı	±	Á	Ñ	á	ñ
2	BPH	PU2	¢	²	Â	Ò	â	ò
3	NBH	STX	£	³	Ã	Ó	ã	ó
4	IND	CCH	¤	´	Ä	Ô	ä	ô
5	NEL	MW	¥	µ	Å	Õ	å	õ
6	SSA	SPA	ı	¶	Æ	Ö	æ	ö
7	ESA	EPA	§	·	Ç	×	ç	÷
8	HTS	SOS	¨	¸	È	Ø	è	ø
9	HTJ	XXX	©	¹	É	Ù	é	ù

Latin-1 Supplement

	018	019	01A	01B	01C	01D	01E
0	b	ε	Ο	υ	ı	İ	Ä
1	B	F	σ	Ο	ı	Ö	ä
2	Б	f	Q	U	†	ø	Æ
3	б	G	q	Y	!	Ů	æ
4	Ь	Y	P	y	DŽ	ů	G
5	ь	hu	p	Z	Dž	Ů	g
6	Ɔ	l	R	z	dž	ů	Č
7	Ć	I	2	3	LJ	Ů	ğ
8	ć	K	z	Č	Lj	ú	Ř

Latin Extended-B



Metadata



Code Point
Repertoire



Variants



Rules

Exercise: “A-G” Language Repertoire

1. The “A-G” language contains following 11 Latin script letters and 1 sign: a, á, b, c, d, e, é, f, f, g, ‡(U+01C2), and ° (U+00B0).

Ask the following questions:

- Is it PVALID?
- Is it used with the language or script in scope?
- Is it suitable in identifiers?
 - Is it in widespread modern use?
 - Is it not technical / religious / limited use only?
 - Is it not really a punctuation / symbol?
 - Is it really necessary for representing identifiers?
- Is the Unicode encoding of the code point stable?
 - Are there any rendering issues?
- How accessible would a TLD containing that code point be?
 - Are there input/keyboard concerns?
- Is it in tension with any of the RFC6912 Principles in any way?

The sign ° (U+00B0) has the property **DISALLOWED**. It cannot be included in the LGR.



Metadata



Code Point
Repertoire



Variants



Rules

Exercise: “A-G” Language Variant Definitions

2. The two forms of ‘f’ (*f*, *f*) are considered “same” by the language community. For example, the label “fin” can be confused with “*f*in”. If both versions are registered to different registrants, it will cause security issues.

Define variants:

- Add variant mappings between *f* (U+0066) and *f* (U+0192) with the type “**blocked**.”



Metadata



Code Point
Repertoire



Variants



Rules

Exercise: “A-G” Language Rule

3. The ≠ must not be at the beginning of any label.

Define Rule for “A-G” language LGR:

Rule #1: ≠ (U+01C2) cannot be at the beginning of a label.



Metadata



Code Point
Repertoire



Variants



Rules



Exercise: Building a “A-G” Language LGR

Metadata

Code Point

Variant

Rule Definition

```
<lgr xmlns="urn:ietf:params:xml:ns:lgr-1.0">
  <meta>
    <version>1</version>
    <unicode-version>15.1.0</unicode-version>
  </meta>
  <data>
    <char cp="0061"/>
    <char cp="0062"/>
    <char cp="0063"/>
    <char cp="0064"/>
    <char cp="0065"/>
    <char cp="0066">
      <var type="blocked" cp="0192"/>
    </char>
    <char cp="0067"/>
    <char cp="00E1"/>
    <char cp="00E9"/>
    <char cp="0192">
      <var type="blocked" cp="0066"/>
    </char>
    <char not-when="at-start-of-word" cp="01C2"/>
  </data>
  <rules>
    <rule name="at-start-of-word">
      <look-behind>
        <start/>
      </look-behind>
      <anchor/>
    </rule>
  </rules>
</lgr>
```

Applying rule to U+01C2

XML

A-G-Language

LGR Version	1
Unicode Version	15.1.0

#	Code Point	Glyph	Script	Name	Tags	Required Context	Variants
1	U+0061	a	Latin	LATIN SMALL LETTER A			
2	U+0062	b	Latin	LATIN SMALL LETTER B			
3	U+0063	c	Latin	LATIN SMALL LETTER C			
4	U+0064	d	Latin	LATIN SMALL LETTER D			
5	U+0065	e	Latin	LATIN SMALL LETTER E			
6	U+0066	f	Latin	LATIN SMALL LETTER F			set 1
7	U+0067	g	Latin	LATIN SMALL LETTER G			
8	U+00E1	á	Latin	LATIN SMALL LETTER A WITH ACUTE			
9	U+00E9	é	Latin	LATIN SMALL LETTER E WITH ACUTE			
10	U+0192	f	Latin	LATIN SMALL LETTER F WITH HOOK			set 1
11	U+01C2	†	Latin	LATIN LETTER ALVEOLAR CLICK		not-when: at-start-of-word	

Variant Set 1 — 2 Members - 1 Mappings

#	Source	Glyph	Target	Glyph	Type(s)	References	Comment
1	U+0066	f	U+0192	f	↔ blocked		

Whole label evaluation and context rules

Name	Regular Expression	Used as Trigger
at-start-of-word	((start))← †	False

HTML



Metadata



Code Point Repertoire



Variants



Rules



Exercise: “A-G” Language Repertoire

Test Label 1: “abé”

✓ VALID

U-label	A-label	Disposition	Code point sequence
abé	xn--ab-cja	valid	U+0061 (a) U+0062 (b) U+00E9 (é)

Test Label 2: “äbé”

✗ INVALID

U-label	A-label	Disposition	Code point sequence
äbé	xn--b-zfaq	invalid	U+00E4 (ä) U+0062 (b) U+00E9 (é)



Metadata



Code Point
Repertoire



Variants



Rules

Exercise: “A-G” Language Variant Definitions

Test Label 3: “*fáce*”

✓ VALID

U-label	A-label	Disposition	Code point sequence
fáce	xn--fce-ela	valid	U+0066 (f) U+00E1 (á) U+0063 (c) U+0065 (e)
Variant Label			
fáce	xn--ce-lia12e	blocked	U+0192 (f) U+00E1 (á) U+0063 (c) U+0065 (e)

Test Label 4: “*fee*”

✓ VALID

U-label	A-label	Disposition	Code point sequence
fee	xn--ee-rxa	valid	U+0192 (f) U+0065 (e) U+0065 (e)
Variant Label			
fee	fee	blocked	U+0066 (f) U+0065 (e) U+0065 (e)



Metadata



Code Point
Repertoire



Variants



Rules

Exercise: “A-G” Language Rule

Test Label 5: “ **cá** ”

✓ **VALID**

U-label	A-label	Disposition	Code point sequence
cá	xn--c-ufa98d	valid	U+0063 (c) U+00E1 (á) U+01C2 ()

Test Label 6: “ **cá** ”

✗ **INVALID**

U-label	A-label	Disposition	Code point sequence
cá	xn--c-ufa78d	invalid	U+01C2 () U+0063 (c) U+00E1 (á)

not-when rule 'at-start-of-word' validates for code point 'U+01C2'
Code point 'U+01C2' does not comply with contextual rules: at-start-of-word
Label 'U+01C2 U+0063 U+00E1' is not in the LGR



Metadata



Code Point
Repertoire



Variants



Rules

Example: Devanagari Label Validation

Devanagari Label: कििताब

✗ INVALID

U-label	A-label	Disposition	Code point sequence
कििताब	xn- -11b4aya5eda	invalid	U+0915 (क) U+093F (ि) U+093F (ि) U+0924 (त) U+093E (ा) U+092C (ब)

Show / hide log

when rule 'follows-C-or-CN' does not validate for code point 'U+093F'
Code point 'U+093F' does not comply with contextual rules: follows-C-or-CN
Label 'U+0915 U+093F U+093F U+0924 U+093E U+092C' is not in the LGR

Example: Devanagari-Gurmukhi Variant Labels

Devanagari
Label

पिढी

U+092A (प) U+093F (ि)
U+0922 (ढ) U+0940 (ी)

Gurmukhi
Label

ਧਿਫੀ

U+0A27 (ਧ) U+0A3F (ਿ)
U+0A2B (ਢ) U+0A40 (ੀ)

Reference LGRs for First Nation Languages in Canada

- Unified Canadian Aboriginal Syllabics (UCAS) Script
- Latin Script

Louis Houle
Co-Chair of UCAS Working Group

Linguistic Overall Context: First Nations & Inuit

- Syllabic Script: 21 Languages



- Latin Script: 80 Languages



Scope of Work

- General Context : LGRs (language and/or script) to be defined in the context of ASCII and French SLD in the same zone example : français.quebec
- Syllabic & Latin Working Group Steps
 - Documenting local language repertoire with Unicode code points
 - Validating the property value for the proposed LGR
 - Attesting the contemporary use
 - Defining in-script variant, cross-script Variant
 - Defining rules
 - Finalize Proposal
 - Review XML-HTML
 - Public Comment
 - Integration

Progress

- Syllabic Script Languages
 - Inuktitut language LGR (covering 7 sub-languages) published 2025
 - Syllabic script LGR **completed**, including Inuktitut, Cree, Ojibway (18 sub-languages) Public Comment expected **March-April 2026**
 - Dene, Dakelh LGR **July-September 2026**
- Latin Script Languages
 - Updated Latin script LGR **completed**, including Nunavut & Labrador roman Inuktitut, Maritimes and Québec (12 sub-languages) Public Comment expected **March-April 2026**
 - Ontario and Manitoba First Nations languages including Cree, Ojibway Iroquian (14 sub-languages) Dene (4 sub-languages) expected **mid-end 2026**
- Remaining LGRs: Syllabic script: 1 language, Latin script: 48 languages

Preventing Security Risk

- Current implementation aims to mitigate the following risks: Homoglyph, Phishing, Spoofing, and Trademark Infringement.
- Two-Level Security Implementation
 - LGR Level: If possible, by blocking or limiting homoglyph situation by restricted usage
 - Registry Level : enforcing bundling mechanism
- Example Latin Script : **le.quebec** -> (é è ê ë **ě ē ě ě ě ě ě**)

ASCIIFrenchFirst Nations

 - The associated domain names i.e., **lé.quebec**, **lè.quebec**, **lê.quebec**, **lě.quebec**, **lē.quebec**, etc. are all allocatable following the same entity principle.
- **Result:** Less room for abuse within the geoTLD scope.
- **Impact:** Safer and better user experience.
- **Public Interest:** Secure DNS.

Multistakeholder Open Model

- Second-Level LGR Panel: UCAS Latin Script Working Group
- 2024 - 2026
 - 26 meetings
 - 25 panel members community members, scholars, linguist, typographer, lexicologist, IT, registry, registry operator, community
- Community Wiki space: [UCAS Script Reference LGR Working Group](#)

Reference LGRs for Indonesian scripts

- Balinese Script
- Javanese Script
- Pegon Script

Gheafany Widyatika Putri
Indonesian Scripts Working Group Member

Why We Started? (Motivation)

- PANDI manages **.id** to **serve Indonesia's Internet community** making the domain space trusted, accessible, and useful for everyone.
- Indonesia has **living local scripts** that remain important for culture, education, and daily community use.
- Internationalized Domain Names (IDNs) allow people to use **their own scripts online**, improving digital inclusion and accessibility.
- To make IDNs **usable and safe**, we need clear, shared rules so domain labels are consistent and not confusing.

Working Group & Scope of Work

- PANDI appoints a community “**champion**” (focal person) who helps:
 - bridge different views within the community,
 - maintain consistency across iterations,
 - and drive the drafting of the Ref. LGR document.



Cokorda Rai Adi Pramatha
Balinese Expert –
Udayana University

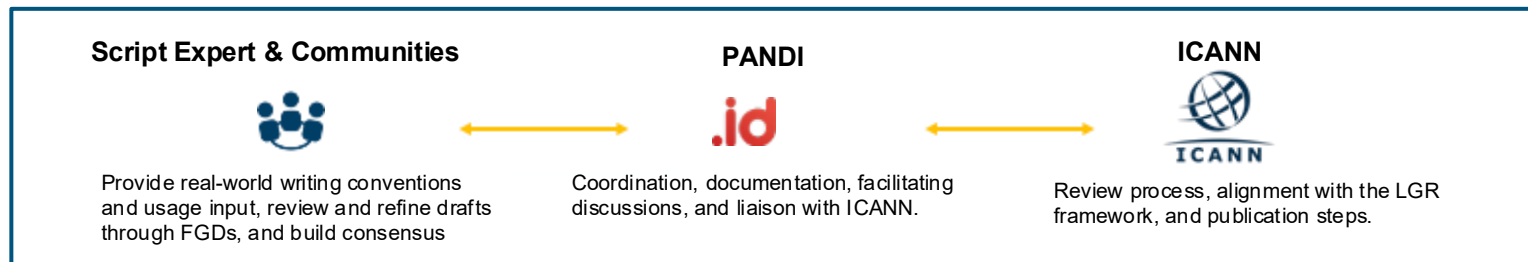


Arif Budiarto
Javanese Expert –
Yogyakarta Special Region Culture Office



Diaz Nawaksara
Pegon Expert –
Script Enthusiasts Community

- Roles



Our Progress

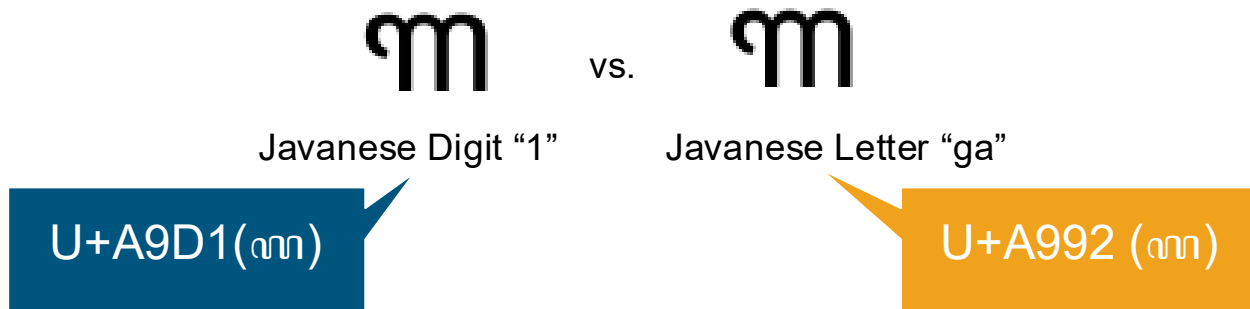
- **Balinese Script**
 - ICANN published the **Balinese Second-Level Reference LGR** in November **2024**.
 - **SLD** (**.ᮘᮞ᮪.id**) has been **launched** with the Governor of Bali in December **2025**.
- **Javanese Script**
 - ICANN visited Yogyakarta to observe Javanese script usage directly with local stakeholders.
 - Javanese Ref. LGR proposal has been submitted and is now in **finalization** with ICANN, moving toward **publication** (leading into the Public Comment stage).
- **Pegon Script**
 - Pegon Ref. LGR proposal has been submitted and is currently awaiting ICANN feedback.



Balancing Usability & Security

- **Security:** We exclude both ASCII and Japanese digits because some digits look similar to Japanese letters (risk of confusion / misuse).
- **Usability:** If users need numbers, we recommend writing them as words (e.g., “one/ siji”) to keep labels readable and consistent.

Example:



What's Next?

- Complete the finalization of the Javanese Ref. LGR and move into the Public Comment stage.
- Address ICANN feedback for the Pegon Ref. LGR through further iterations with the community.
- Strengthen adoption: outreach materials, practical examples, and guidance for broader stakeholders.
- Continue roadmap work for other Indonesian scripts (such as Batak, Buginese, Sundanese) based on community readiness and demand.

Thank You



icann.org



[@icann](https://twitter.com/icann)



facebook.com/icannorg



youtube.com/icannnews



flickr.com/icann



linkedin.com/company/icann



instagram.com/icannorg