



87

ANNUAL
GENERAL
MEETING



How It Works

Internationalized Domain Names

20 October 2026

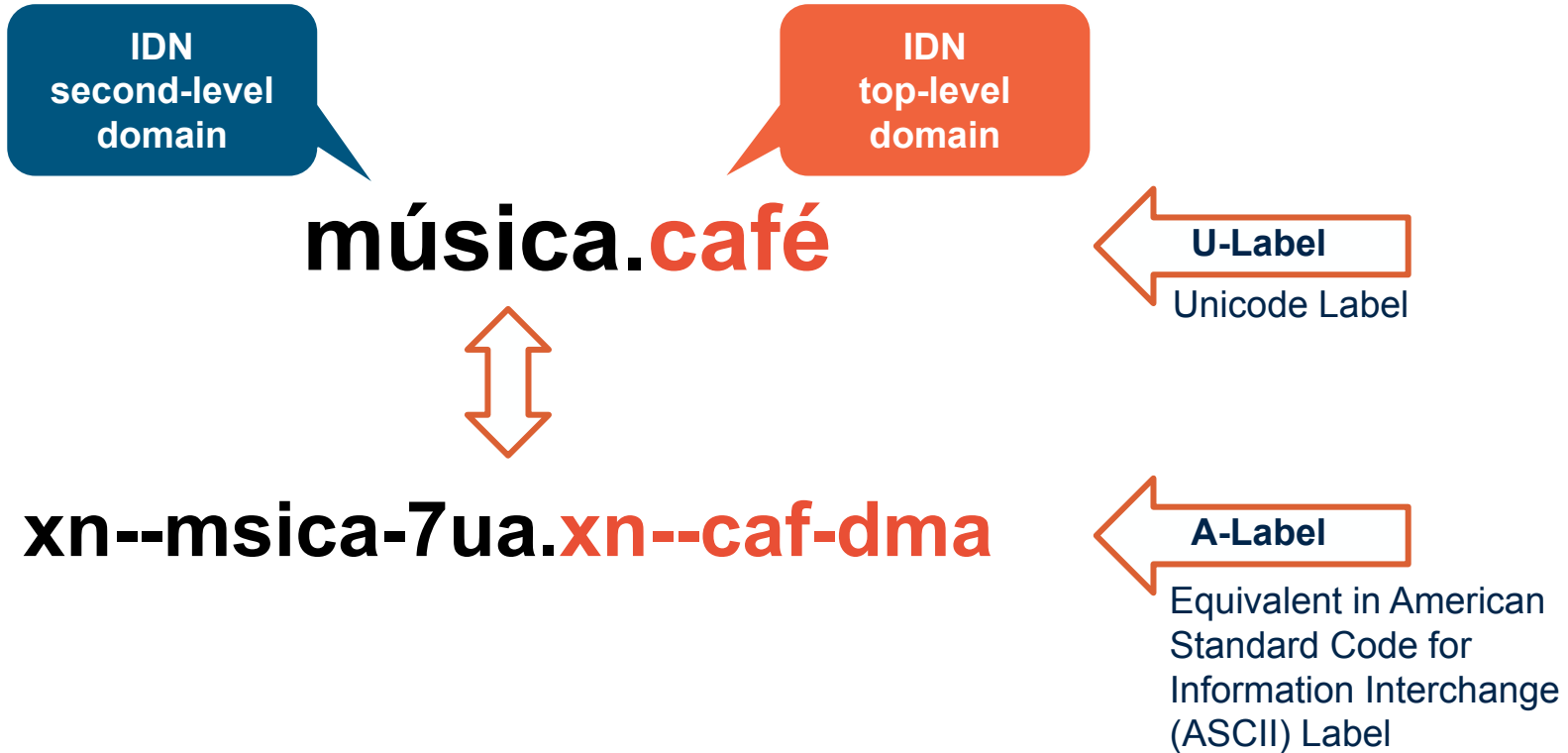
Agenda

- Enabling Internationalized Domain Names (IDNs) in local languages and scripts
 - The process to develop the Label Generation Rules (LGRs)
 - Need for Community-Driven Development of Rules for IDNs
- Reference LGRs for scripts used in Indonesia
 - Overview of scripts used for IDNs in Indonesia
 - Balinese script
 - Javanese script
 - Arabic script including Pegon characters
- Q&A (10 mins)

The Process to Enable IDNs in Local Languages and Scripts

Pitinan Kooarmornpatana
IDN Program Director
ICANN

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 IDN labels used in ICANN registry contexts must satisfy applicable IDNA2008 requirements

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

May be used only when
specified contextual
requirements are met

Example:

“l.l”

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

(U+00B7 must occur
between two lowercase ‘l’
characters (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+05CA -
U+05CF

*The fact that a label consists only of PVALID code points does not mean that it can necessarily be used in the DNS.

Additional Requirements Beyond 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...”*
- **Because additional restrictions beyond IDNA2008 are necessary, registries use Label Generation Rules (LGRs) to define them.**

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

What are code points to be used?

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 addresses
 - 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 form a word in a language
 - Not restricted to “correct” spellings
 - May not carry a meaning in the “lexical” sense

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

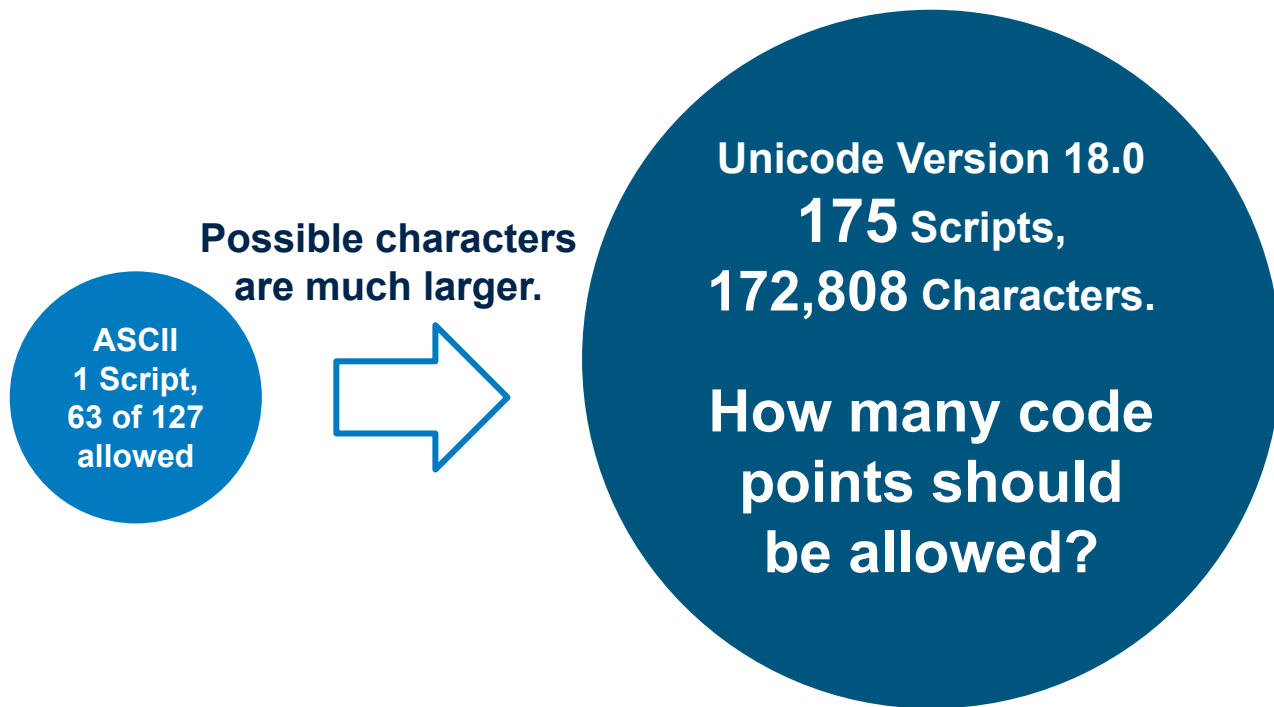
Domain Name Mnemonics in ASCII

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

ASCII Printable Characters (Hex Code)

	2	3	4	5	6	7
0	space	0	@	P	`	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	\$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	
D	-	=	M	]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	del

ASCII vs Unicode



Example of Arabic Shortlisted Code Points

- Based on the [Proposal for Arabic Script Root Zone Label Generation Rules](#)
- Yellow-highlighted code points are included in the repertoire

060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F	075	076	077	08A	08B	08C	08D	08E	08F	
0	ي	ذ	-	ـ	٠	١	٢	٣	٤	٥	٦	٧	٨	٩	٠	0	ي	ي	ش	ب					٠
1	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤	1	١٠	١١	١٢	١٣					١
2	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠	٣١	٣٢	٣٣	٣٤	٣٥	٣٦	٣٧	٣٨	٣٩	2	٢٥	٢٦	٢٧	٢٨					٢
3	٤٠	٤١	٤٢	٤٣	٤٤	٤٥	٤٦	٤٧	٤٨	٤٩	٥٠	٥١	٥٢	٥٣	٥٤	3	٤٠	٤١	٤٢	٤٣					٣
4	٥٥	٥٦	٥٧	٥٨	٥٩	٦٠	٦١	٦٢	٦٣	٦٤	٦٥	٦٦	٦٧	٦٨	٦٩	4	٥٥	٥٦	٥٧	٥٨					٤
5	٧٠	٧١	٧٢	٧٣	٧٤	٧٥	٧٦	٧٧	٧٨	٧٩	٨٠	٨١	٨٢	٨٣	٨٤	5	٧٠	٧١	٧٢	٧٣					٥
6	٨٥	٨٦	٨٧	٨٨	٨٩	٩٠	٩١	٩٢	٩٣	٩٤	٩٥	٩٦	٩٧	٩٨	٩٩	6	٨٥	٨٦	٨٧	٨٨					٦
7	١٠٠	١٠١	١٠٢	١٠٣	١٠٤	١٠٥	١٠٦	١٠٧	١٠٨	١٠٩	١١٠	١١١	١١٢	١١٣	١١٤	7	١٠٠	١٠١	١٠٢	١٠٣					٧
8	١١٥	١١٦	١١٧	١١٨	١١٩	١٢٠	١٢١	١٢٢	١٢٣	١٢٤	١٢٥	١٢٦	١٢٧	١٢٨	١٢٩	8	١١٥	١١٦	١١٧	١١٨					٨
9	١٣٠	١٣١	١٣٢	١٣٣	١٣٤	١٣٥	١٣٦	١٣٧	١٣٨	١٣٩	١٤٠	١٤١	١٤٢	١٤٣	١٤٤	9	١٣٠	١٣١	١٣٢	١٣٣					٩
A	١٤٥	١٤٦	١٤٧	١٤٨	١٤٩	١٥٠	١٥١	١٥٢	١٥٣	١٥٤	١٥٥	١٥٦	١٥٧	١٥٨	١٥٩	A	١٤٥	١٤٦	١٤٧	١٤٨					١٠
B	١٦٠	١٦١	١٦٢	١٦٣	١٦٤	١٦٥	١٦٦	١٦٧	١٦٨	١٦٩	١٧٠	١٧١	١٧٢	١٧٣	١٧٤	B	١٦٠	١٦١	١٦٢	١٦٣					١١
C	١٧٥	١٧٦	١٧٧	١٧٨	١٧٩	١٨٠	١٨١	١٨٢	١٨٣	١٨٤	١٨٥	١٨٦	١٨٧	١٨٨	١٨٩	C	١٧٥	١٧٦	١٧٧	١٧٨					١٢
D	١٩٠	١٩١	١٩٢	١٩٣	١٩٤	١٩٥	١٩٦	١٩٧	١٩٨	١٩٩	٢٠٠	٢٠١	٢٠٢	٢٠٣	٢٠٤	D	١٩٠	١٩١	١٩٢	١٩٣					١٣
E	٢٠٥	٢٠٦	٢٠٧	٢٠٨	٢٠٩	٢١٠	٢١١	٢١٢	٢١٣	٢١٤	٢١٥	٢١٦	٢١٧	٢١٨	٢١٩	E	٢٠٥	٢٠٦	٢٠٧	٢٠٨					١٤
F	٢٢٠	٢٢١	٢٢٢	٢٢٣	٢٢٤	٢٢٥	٢٢٦	٢٢٧	٢٢٨	٢٢٩	٢٣٠	٢٣١	٢٣٢	٢٣٣	٢٣٤	F	٢٢٠	٢٢١	٢٢٢	٢٢٣					١٥

Building the Code Point Repertoire

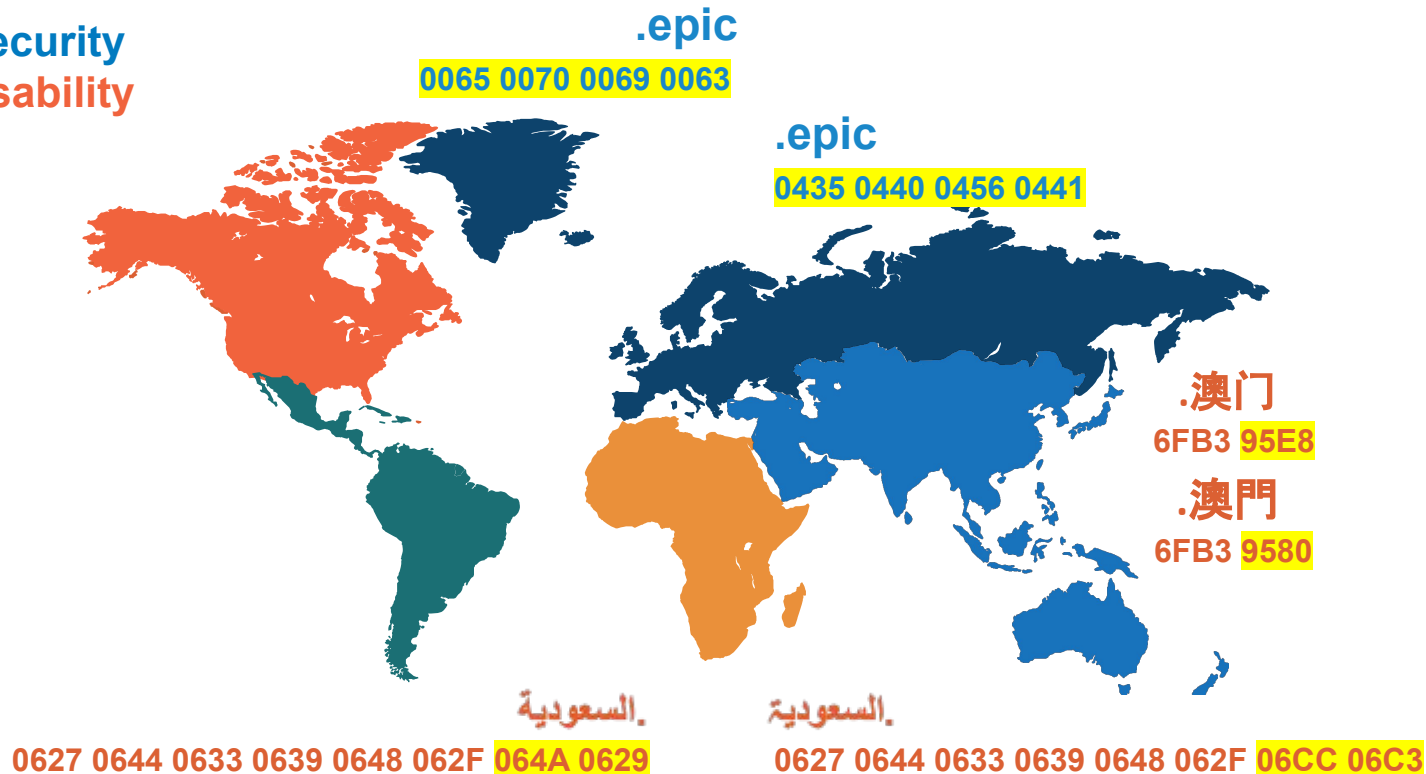
Before including a code point, ask the following questions:

- Is it PVALID?
- Is it used with the language or script in scope?
- Is it suitable in identifiers?
 - Is it in widespread modern use?
 - Does it function primarily as punctuation or a symbol rather than as part of ordinary written identifiers?
 - 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 RFC 6912 Principles in any way?

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

Understanding Variants

- Security
- Usability



Variants - What is the Goal?

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

Defining Variants

- Would a knowledgeable user familiar with 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?
- Are there any cross-script variants?

Examples of Cross-Script Variants

Cyrillic	Greek	Latin
а U+0430 CYRILLIC SMALL LETTER A	α U+03B1 GREEK SMALL LETTER ALPHA	a U+0061 LATIN SMALL LETTER A
о U+043E CYRILLIC SMALL LETTER O	ο U+03BF GREEK SMALL LETTER OMICRON	o U+006F LATIN SMALL LETTER O
р U+0440 CYRILLIC SMALL LETTER ER	ρ U+03C1 GREEK SMALL LETTER RHO	p U+0070 LATIN SMALL LETTER P

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

Rules - What is the Goal?

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

Example of Risk Mitigation (Thai Script)



- 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 from following another U+0E40

Example of Risk Mitigation (Lao Script)

“U+0EAE (ສ) U+0EC9(ັ) U+0EC9(ັ)”

Consonant

Tone Mark

Tone Mark



Chrome on Android 16



Chrome on iOS 26.3

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

Root Zone LGR for script: Arabic (Arab)

Date	2025-09-23
LGR Version	6 (Root Zone LGR for the Arabic Script)
Language	und-Arab (Arabic Script)
Scope	domain: "*" (Root)
Unicode Version	16.0.0

rz-lgr-6-arabic-script-23sep25-en

This document is mechanically formatted from the above XML file for the LGR. It provides additional summary data and explanatory text. The XML file remains the sole normative specification of the LGR.

Table of Contents

1 Description

2 Repertoire

3 Variants

4 Classes, Rules and Actions

4.1 Character Classes

4.2 Whole Label Evaluation and Context Rules

4.3 Actions

5 Table of References

1 Description

Root Zone Label Generation Rules for the Arabic Script

Overview

This file contains a set of Label Generation Rules (LGR) for the Arabic script for the Root Zone. For more details on this LGR and additional background on the script, see TF-AIDN, "Proposal for Arabic Script Root Zone LGR", Version 3.4, 18 November 2024.

one of a set of LGR files that together form an integrated LGR for the DNS Root Zone [RZ-LGR-61].

Repertoire

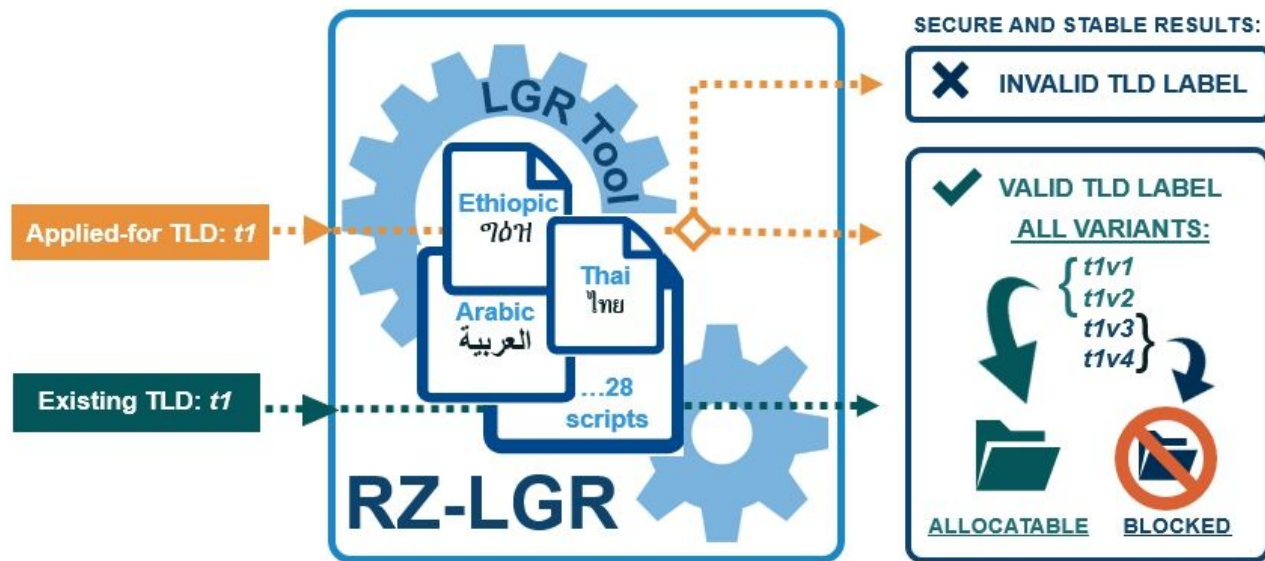
- 

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.
 - The User Guide is 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](#).

Usage of LGR

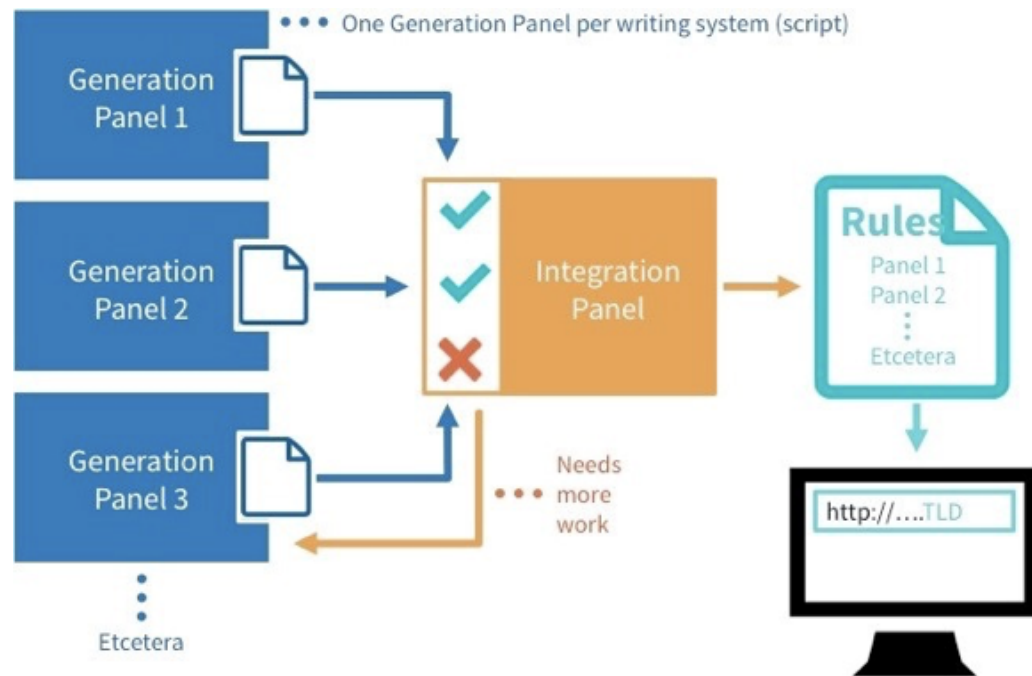
- An LGR in machine-readable format can be processed by the LGR Tool to validate a label and generate or validate its variant labels.
- For example: Root Zone Label Generation Rules



Need for Community-Driven Development of Rules for IDNs

Root Zone Label Generation Rules (RZ-LGR)

- Multistakeholder model in action in the Root Zone Label Generation Rules (RZ-LGR) Project, as per RZ-LGR Procedure
- RZ-LGR Version 6 supports 27 scripts which covers more than 387 languages.
- 11 years of work, 18 Generation Panels from 45 countries.
- Over 270 script community volunteers contributing more than 10,000 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 to 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 covering 62 languages and scripts:

- **30 scripts:** Arabic, Armenian, Balinese, Bangla (Bengali), Chinese, Cyrillic, Devanagari, Ethiopic, Georgian, Greek, Gujarati, Gurmukhi, Hebrew, Japanese (Hiragana, Katakana, Kanji [Han]), Javanese, Kannada, Khmer, Korean (Hangul, Hanja [Han]), Lao, Latin, Malayalam, Myanmar, Oriya, Sinhala, Tamil, Telugu, Thaana, Thai, and Unified Canadian Aboriginal Syllabics (UCAS).
- **32 languages:** 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.

Collaboration on Developing the Ref. LGR

Voluntary expertise provided by a community working group

- Linguistic expertise to develop a proposal that defines repertoire, variants, and rules.
- Word lists or corpora for languages written in the script.
- Review and test the LGR.

Support provided by ICANN org

- Training for the community on designing an LGR.
- 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 publishing an LGR.



Reference LGRs for Scripts Used in Indonesia

Hammam Riza - Chairman of the Supervisory Board of PANDI

Cokorda Pramatha, PhD - Balinese Script Expert

Arif Budiarto - Javanese Script Expert

Diaz Nawaksara - Arabic Script Expert supporting Pegon

Indonesia's Journey Toward IDNs in Local Scripts



Balinese

Reference LGR →
.id implementation



Javanese

Reference LGR →
.id implementation



Pegon

Reference LGR →
.id implementation &
prepare RZ-LGR
proposal

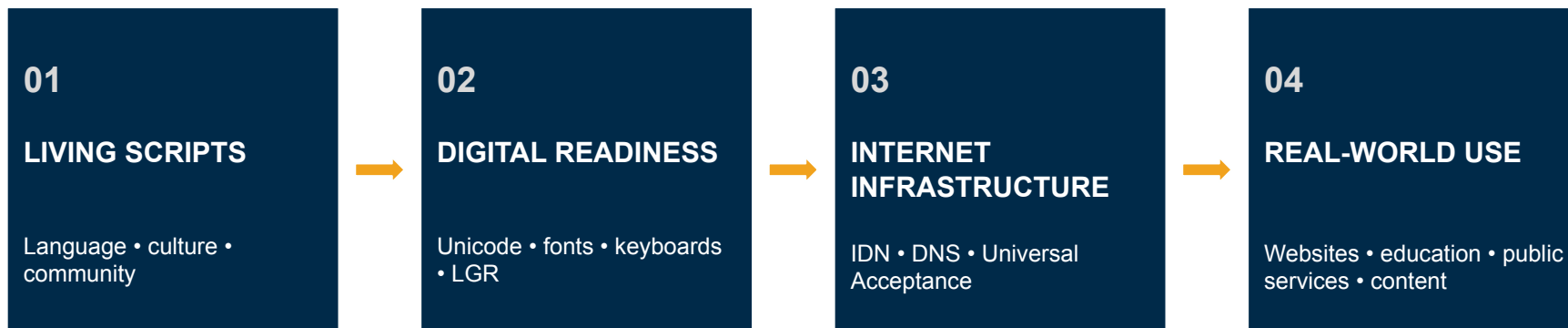


Next Scripts

Buginese • Batak •
Sundanese • others

**For Indonesia, IDNs are not only about displaying local scripts in domain names.
They are about enabling our writing systems to function safely and meaningfully on the Internet**

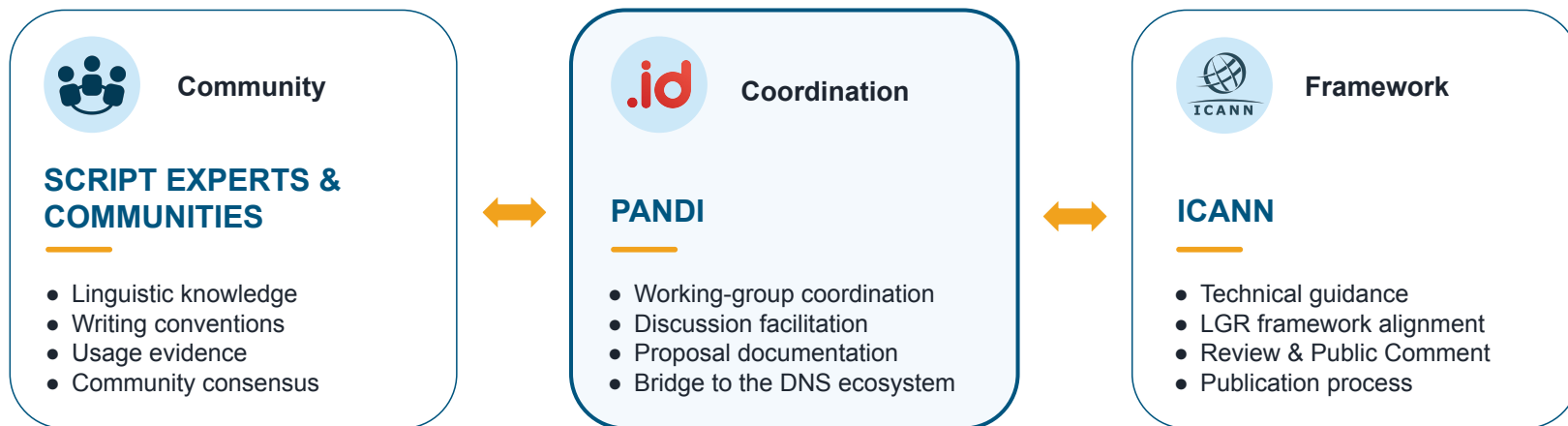
From Cultural Heritage to Digital Use



Preserve → Standardize → Enable → Use

The objective is not simply to preserve local scripts, but to make them usable in today's digital ecosystem

Community-Driven Development in Indonesia



“The rules should come from the community that actually uses the script”

Balinese script

Balinese second-level domains: live under .id

ကဏ္ဍကိ. id

၇၇၇၇ is the second-level label. .id is the top-level domain.

a-label form: xn--9tfky.id

1. Community and Purpose

Balinese-script domains give communities a digital identity in their own writing system.

Potential uses include learning resources and cultural archives.

Community script expertise guides the rules.

2. Published LGR

ICANN published the reference LGR in November 2024.

65 Balinese letters and signs, plus hyphen-minus.

Digits are excluded. Equivalent variants are blocked. Context rules validate character order.

3. Live implementation and next steps

Launched: 11 December 2025
PANDI: registry Universitas
Udayana: dedicated registrar
bali.domain.id

98 unique domains registered

Next: email infrastructure and UA testing of links, forms and email.

Javanese script

1. Cultural Demand

Widespread daily adoption across Central Java, East Java, and Yogyakarta requires native digital domain identity.

80M+ Native Speakers

Active regional language base in Indonesia.

Education & Governance

Mandatory school curricula & official public signage.

Digital Media & Commerce

Javanese Wikipedia, mobile apps, and SME branding.

2. Technical & Security LGR Details

1. Excluded Punctuation & Archaic

Historical symbols and obsolete characters excluded from IDN scope.

Example:



U+A985
JAVANESE LETTER I KAWO



U+A9DF
JAVANESE PADA ISEN-ISEN

2. Excluded Digits (Homograph Threat)

Omitted due to identical visual forms with *Nglegena* consonants.

Example:



U+A9D7
JAVANESE DIGIT SEVEN



U+A9AD
JAVANESE LETTER LA

3. Included Aksara Rekan (Loanwords)

Retained for contemporary foreign loanwords and personal names.

Example:



FA
U+A9A5 U+A9B3



FAJAR
U+A9A5 U+A9B3 U+A997 U+A982

4. Blocked Aksara Murda Variants

Mapped and blocked as variants due to identical sound and meaning.

Example:



SINTA
U+A9AF U+A9B6
U+A9A4 U+A9C0 U+A9A0



SINTA
U+A9B0 U+A9B6
U+A9A4 U+A9C0 U+A9A0

5. Prohibited 3-Stacked Conjuncts

Below-2-level stacks prohibited due to browser address bar rendering limits.

6. Strict WLE Validation Rules

Whole-Label Evaluation for *Sandhangan*, *Pasangan* & *Pangkon* sequencing.

3. What Next?

1 Phase-1

Implement Second-Level Domains (SLD) with Javanese script under .id
e.g.,
[domain].ꦗꦮꦤ꧀.id

2 Phase-2

Full Internationalized Domain Name (IDN) top-level domain deployment.
e.g.,
[domain].ꦲꦩꦧꦤ꧀ꦠꦺꦴꦩꦤ꧀

Arabic Script with Pegon Characters

Toward a Broader “Nusantara Arabic Script” Framework

1 Community & Script Scope

Indonesia has a living tradition of writing local languages in Arabic script. Pegon is one such tradition, used for Javanese, Sundanese, and Madurese. We will adopt the term “Nusantara Arabic Script” to include other Arabic-script traditions across Indonesia.

Inclusive Terminology

Our scope is expanded to “Nusantara Arabic Script” covering Arabic-script traditions including:

Jawi
(Melayu)

ملايو

Serang
(Bugis)

سيرڠ

Buri
Wolio
(Wolio)

بُرُولِي

Pegon
(Javanese,
Sundanese,
Madurese, etc.)

ڦيڦون



This approach reflects the living use of Arabic-derived scripts across Indonesian linguistic communities.

2 LGR Approach: Arabic-Based, Locally Inclusive

Based on the existing Arabic script reference LGR framework, identify code points required by Indonesian Arabic-script communities.



Key Unicode Characters

(addition to existing Arabic script LGR)

ط

U+088C

ARABIC LETTER TAH WITH
THREE DOTS BELOW
(Pegon)

ڦ

U+08AE

ARABIC LETTER DAL WITH
THREE DOTS BELOW
(Pegon)



Arabic Extended-B (U+088C)

Encoded in Unicode 14.0 (September 2021)



Security & Usability

- **Legacy Devices:** Older systems may lack support for these newer Unicode characters.
- **Pegon Usability:** Font rendering, text direct, vary across devices, requiring further testing (RTL script)

3 IDN Example & Next Steps

This example illustrates the use of Pegon in an IDN label. Our next steps focus on technical validation, community participation, and readiness for wider use..

ڦاندي.ڦيڦون

www.pandi.pegon

Next Steps

- 1 Continue developing and testing repertoire, variants, and rules with Indonesian script communities.
- 2 Consolidate nationwide outreach to strengthen IDN user potential in future.
- 3 Coordinate with relevant stakeholders to align technical, linguistic, and policy aspects of RZ-LGR development
- 4 Collect field evidence of previously undocumented Arabic-script usage in Indonesia.
- 5 Prepare registrars to validate Arabic-script domain labels across local languages.

ICANN Webpage and Social Media Links



icann.org



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



facebook.com/icannorg



youtube.com/icannnews



flickr.com/icann



linkedin.com/company/icann



instagram.com/icannorg