

# IDN Tables for the Second Level Domain Names



Sarmad Hussain  
Senior Director, IDN and UA Programs

ICANN 74  
15 June 2022

**IDN Table**  
**= Variant Table**  
**= Label Generation Rules (LGR)**

# Rules for ASCII Domain Name Labels

**www.cafe-123.com**



2

## Forming ASCII Labels

Use LDH

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

Repertoire

Rules

Label length = 63

Other constraints (e.g., on hyphen)

1

## Forming ASCII Labels

Use only Letters

- Letters [a-z]
- Label length = 63

# Domain Name Mnemonics in ASCII

Using LDH

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

2

	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

# Top-level Domain Name Mnemonics in ASCII

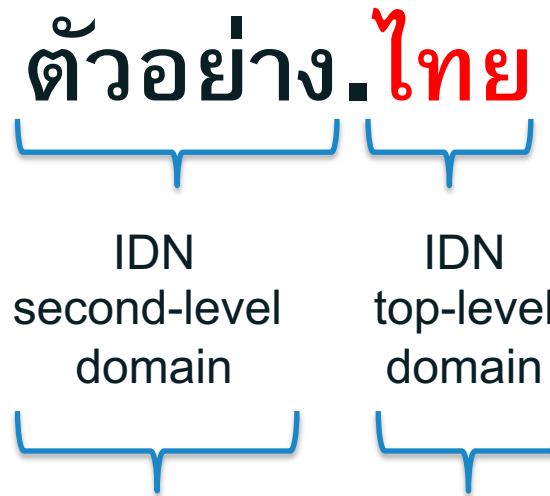
Using Letters only

- Letters [a-z]
- ~~Digits [0-9]~~
- ~~Hyphen (-)~~

1

	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

# Internationalized Domain Name (IDN) Labels



## Syntax of IDN Labels

**Valid U-Label:** Unicode code points as constrained by the “LDH” scheme within IDNA 2008

## Syntax of IDN Labels

**Valid U-label,** further constrained by the “letter” principle for TLDs

② = IDN Table or LGR

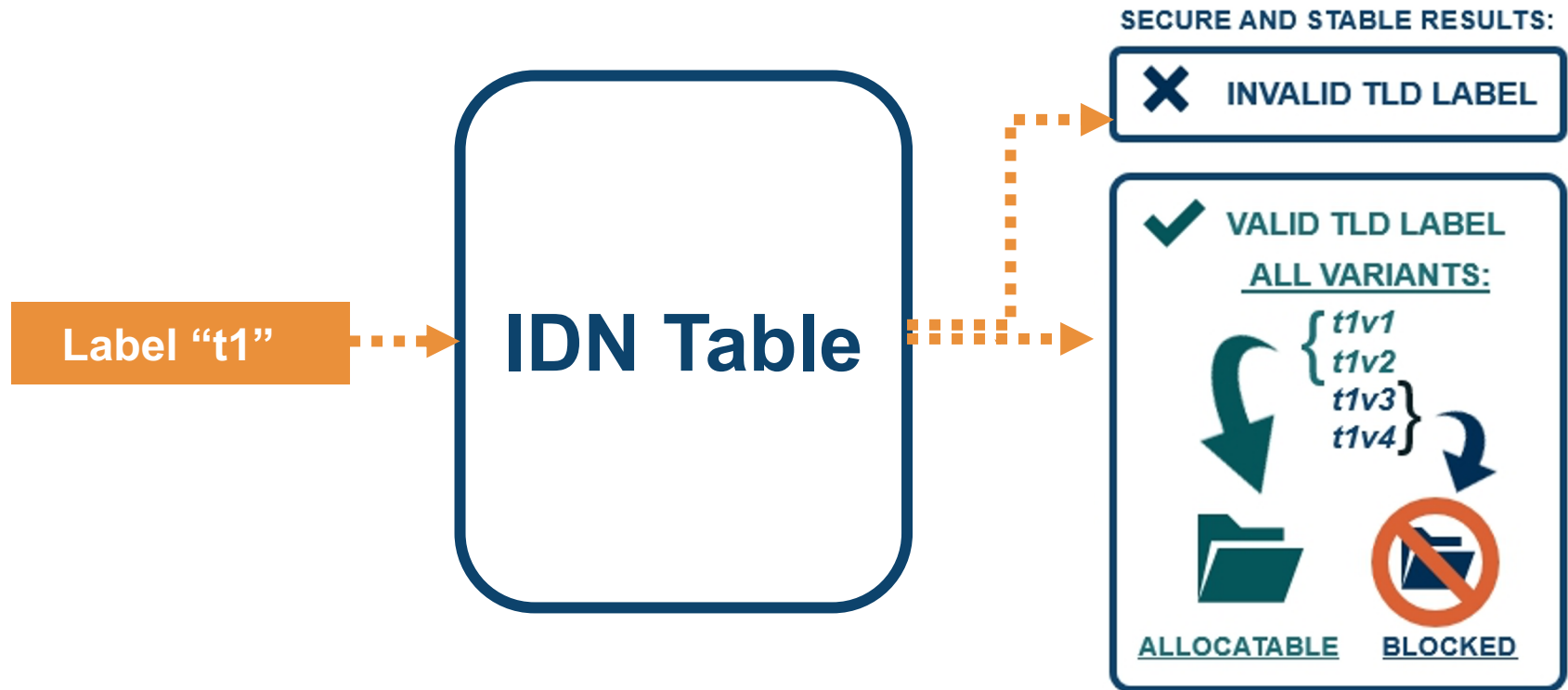
① = RZ-LGR

# Objective of IDN Tables

---

Enable second-level domain names  
**in the local languages and scripts**  
used by the communities globally  
**in a secure and stable manner**

# What Does the IDN Table Do?



1. Validate a label, and
2. Generate the allocatable or blocked variant labels of a valid label.



# What is an IDN Table?

- IDN tables define which labels can be registered for a particular language or script at the second-level under a top-level domain (TLD).
  - Manage security, stability, and usability of IDNs.
  
- IDN tables include Unicode code points, variants as well as linguistic and technical constraints to determine appropriate and secure domain labels.
  1. **Language and script tag**
  2. **Metadata/Description**
  3. **Repertoire**: {U+0061 - U+007A, U+0030 - U+0039, U+002D}; a-z, 0-9, -.
  4. **Variants**: U+0629 (ّ) and U+06C3 (ّ) are variants in Arabic script.
  5. **Rules**:
    - Linguistic: A tone mark follows the main consonant in Lao script.
    - Technical: The Unicode string **MUST NOT** contain Hyphen in the third- and fourth-character positions as per RFC5891.
  
- IDN table can be defined in multiple formats:
  - Latest machine-readable XML format specified in RFC 7940.
    - RFC 7940 refers to IDN table as Label Generation Rules (LGR).
  - Earlier, text-based formats in RFC 3743 and RFC 4290 are also used.

# Evolution of IDN Table Formats

---

- ⦿ [RFC3743](#) (2004) - Informational
  - Encoded in TXT format.
  - List of code points. Variants for code points using semi-colons (;).
  - Rules in the comment section, not machine-readable.
- ⦿ [RFC4290](#) (2005) - Informational
  - Encoded in TXT format.
  - List of code points. Variants for code point using pipe symbol (|).
  - Rules in the comment section, not machine-readable.
- ⦿ [RFC7940](#) (2016) – Standards Track
  - Encoded in XML format.
  - List of code points, variants of a code point, and rules are machine-readable.

# Example of IDN Table in RFC3743 Format

```
# Script/Language Identifier: zh
# Script/Language Description: Chinese
# Version: 1.1
# Effective Date: 2012-03-29
# Registry: EXAMPLE
# Code points: U+002D (HYPHEN-MINUS) Reference: RFC 5891 (sec 4.2.3.1)
# Rules: Label must neither start nor end with U+002D.
# Label must not have U+002D in both third and fourth position.
#
# This table conforms to the format specified in RFC 3743.
#
U+002D;U+002D;U+002D # HYPHEN-MINUS
U+0030;U+0030;U+0030 # DIGIT ZERO
U+0031;U+0031;U+0031 # DIGIT ONE
U+0032;U+0032;U+0032 # DIGIT TWO
U+0033;U+0033;U+0033 # DIGIT THREE
U+0034;U+0034;U+0034 # DIGIT FOUR
U+0035;U+0035;U+0035 # DIGIT FIVE
U+0036;U+0036;U+0036 # DIGIT SIX
U+0037;U+0037;U+0037 # DIGIT SEVEN
U+0038;U+0038;U+0038 # DIGIT EIGHT
U+0039;U+0039;U+0039 # DIGIT NINE
U+3447;U+3447;U+3447,U+3473 # <CJK Ideograph Extension A>
U+3473;U+3447;U+3447,U+3473 # <CJK Ideograph Extension A>
U+359E;U+359E;U+359E,U+558E # <CJK Ideograph Extension A>
U+360E;U+360E;U+360E,U+361A # <CJK Ideograph Extension A>
U+361A;U+360E;U+360E,U+361A # <CJK Ideograph Extension A>
U+3918;U+3918;U+3918,U+396E # <CJK Ideograph Extension A>
U+396E;U+3918;U+3918,U+396E # <CJK Ideograph Extension A>
U+39CF;U+39CF,U+6386 #
```

Rules are described in the comment section.

Code point and variant definition are listed in tabular text format using ‘;’.


# Example of IDN Table in RFC4290 Format

```
#Arabic IDN Language table
#Language Tag: ar
#Language Description: Arabic
#Version: 2.1
#Effective Date: December 6, 2017
#Registry: EXAMPLE
# Notes: This table describes code points allowed for the Arabic script.
# Registrations using either Arabic-Indic or Arabic digits are allowed,
# but digit types are not allowed to be mixed and cannot be at the
# beginning of a label.

#Code points

U+002D
U+0030|U+0660
U+0031|U+0661
U+0032|U+0662
U+0033|U+0663
U+0034|U+0664
U+0035|U+0665
U+0036|U+0666
U+0037|U+0667
U+0038|U+0668
U+0039|U+0669
U+0621
U+0622|U+0623
```

Rules are described in the comment section.

Code point and variant definition are listed in tabular text format using '|'.  


# Example of IDN Table in RFC7940 Format

```
<lgr xmlns="urn:ietf:params:xml:ns:lgr-1.0">
<meta>
<version comment="IDN Reference table for Arabic language">4.1</version>
<date>2021-11-24</date>
<language>ar</language>
<unicode-version>6.3.0</unicode-version>
<description type="text/plain">
<![CDATA[ #Table format and IDNA 2008 Language table is in RFC 7940 format
#Contact Info Contact Name: EXAMPLE ]]>
</description>
</meta>
<data>
<char cp="002D" not-when="hyphen-rules"/>
<char cp="0030" not-when="no-digit-at-start">
<var cp="0660" type="blocked"/>
</char>
<char cp="0031" not-when="no-digit-at-start">
<var cp="0661" type="blocked"/>
</char>
<char cp="0032" not-when="no-digit-at-start">
<var cp="0662" type="blocked"/>
</char>
<char cp="0033" not-when="no-digit-at-start">
<var cp="0663" type="blocked"/>
</char>
<char cp="0034" not-when="no-digit-at-start">
<var cp="0664" type="blocked"/>
</char>
<char cp="0035" not-when="no-digit-at-start">
<var cp="0665" type="blocked"/>
</char>
<char cp="0036" not-when="no-digit-at-start">
<var cp="0666" type="blocked"/>
```

← List of code points, variant definition on each applicable code point, rules are machine-readable.

# Why the XML Format?

---

- ⦿ Rules are more precise than writing them in plain English.
  - Prevents mis-interpretation and promotes consistent use of rules by different parties.
- ⦿ The LGR can be developed, reviewed, compared, etc. automatically by tools and manual inspection is not needed.
  - Tools to design, create and process the XML LGRs available openly by ICANN org.
- ⦿ Straightforward to test the LGR on labels automatically.
- ⦿ Allows flexibility in encoding as the text-based formats – as shown in encoding 26 scripts for the RZ-LGR.
- ⦿ Straightforward to create human readable form in HTML, automatically derived from the XML data.

# Who Develops the IDN Tables?

---

- ⊙ IDN Tables for the second level:
  - Developed and used by the registries for the second level.
  - Registries may refer to the Reference LGR when developing their IDN Table.
  - IDN table should not have any security and stability problems.
  
- ⊙ RZ-LGR for the root zone:
  - Developed by script communities and anticipated for use by ICANN for the root zone.
  - Developed the RZ-LGR as per the [LGR Procedure](#).

# Example: Mini Latin Script IDN Table

## Example IDN Table

### List of code points

Code Point	Glyph	Name
U+002D	-	MINUS-HYPHEN
U+0062	b	LATIN SMALL LETTER B
U+0063	c	LATIN SMALL LETTER C
U+0069	i	LATIN SMALL LETTER I
U+0131	ı	LATIN SMALL LETTER DOTLESS I

### Variant definition

0069(i) -> 0131(i) blocked  
0131(i) -> 0069(i) allocatable

### Rules

Hyphen cannot be at the first position of a label

“bbc”

“bbc” ✓

“cio”

“cio” ✗

‘o’ is not in the code point repertoire

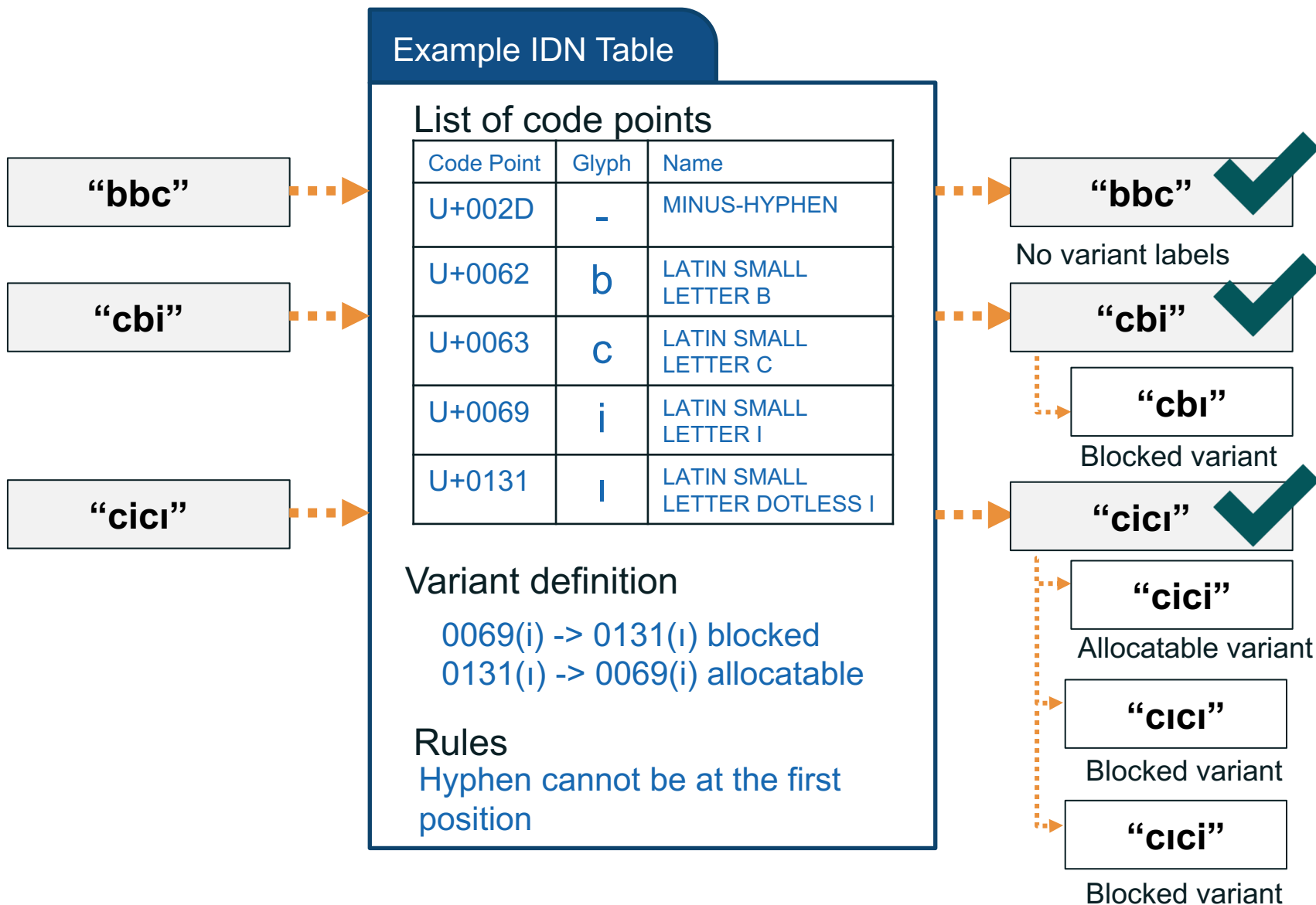
“-bbb”

“-bbb” ✗

Hyphen cannot be at the first position of a label.



# Example: Mini Latin Script IDN Table



# Activating Variant Labels at the Second Level

---

- ⦿ The IDN tables may generate allocatable and blocked variant labels.
- ⦿ Allocatable variant labels can be activated if allowed by the registry policy and requested by the registrant.
- ⦿ [IDN Guidelines version 4.0](#) suggest a mechanism for automatically activating variants by the registry without the request from the registrant for scripts identified by the community.
- ⦿ IDN Guidelines version 4.0 suggest all such activated variant labels should be registered to the same registrant (same entity) to prevent user confusion.

# Harmonizing Multiple IDN Tables Under TLDs

---

- ⦿ A TLD can implement more than one IDN tables.
- ⦿ IDN tables under a TLD should be harmonized ([IDN Guidelines version 4.0](#)).
- ⦿ IDN tables under a TLD and its variant TLDs should also be harmonized ([IDN Variant TLD Recommendations](#)).

# Harmonizing Multiple IDN Tables Under TLDs

---

- ⦿ IDN Guidelines version 4.0 explain harmonization as two measures. These two measures are suggested to prevent cases of IDN variant labels being generated by different IDN Tables under the same TLD to be allocated to different registrants.
  - i. two IDN variant code points or IDN variant code point sequences in one IDN Table cannot be non-IDN-variant code points or non-IDN-variant code point sequences in another IDN Table implemented under the same TLD, and
  - ii. all code points in all the IDN Tables under the same TLD must be collectively considered for analysis of IDN variants of code points for each of these IDN Tables. This can be achieved by developing a “Merged LGR” from all element LGRs.

# Example: The Need of Harmonization for IDN Tables

- The two variant (visually identical) labels are created as non-variant labels under the two different IDN tables: **بيف.TLD** and **بيف.TLD**.

## Arabic Language IDN Table

### List of code points

Code Point	Glyph	Name
U+0628	ب	ARABIC LETTER BEH
U+0641	ف	ARABIC LETTER FEH
U+064A	ي	ARABIC LETTER YEH

“بيف”

U+0628 U+064A  
U+0641

“بيف” ✓

U+0628 U+064A  
U+0641  
No variant labels

## Urdu Language IDN Table

### List of code points

Code Point	Glyph	Name
U+0628	ب	ARABIC LETTER BEH
U+0641	ف	ARABIC LETTER FEH
U+06CC	ی	ARABIC LETTER FARSI YEH

“بيف”

U+0628 U+06CC  
U+0641

“بيف” ✓

No variant labels

# Example: The Need of Harmonization for IDN Tables

- If a TLD offers both Arabic and Urdu languages, the IDN tables harmonized to identify variant labels across the IDN tables.

Merged IDN Table

List of code points

Code Point	Glyph	Name
U+0628	ب	ARABIC LETTER BEH
U+0641	ف	ARABIC LETTER FEH
U+064A	ي	ARABIC LETTER YEH
U+06CC	ى	ARABIC LETTER FARSI YEH

Variant definition

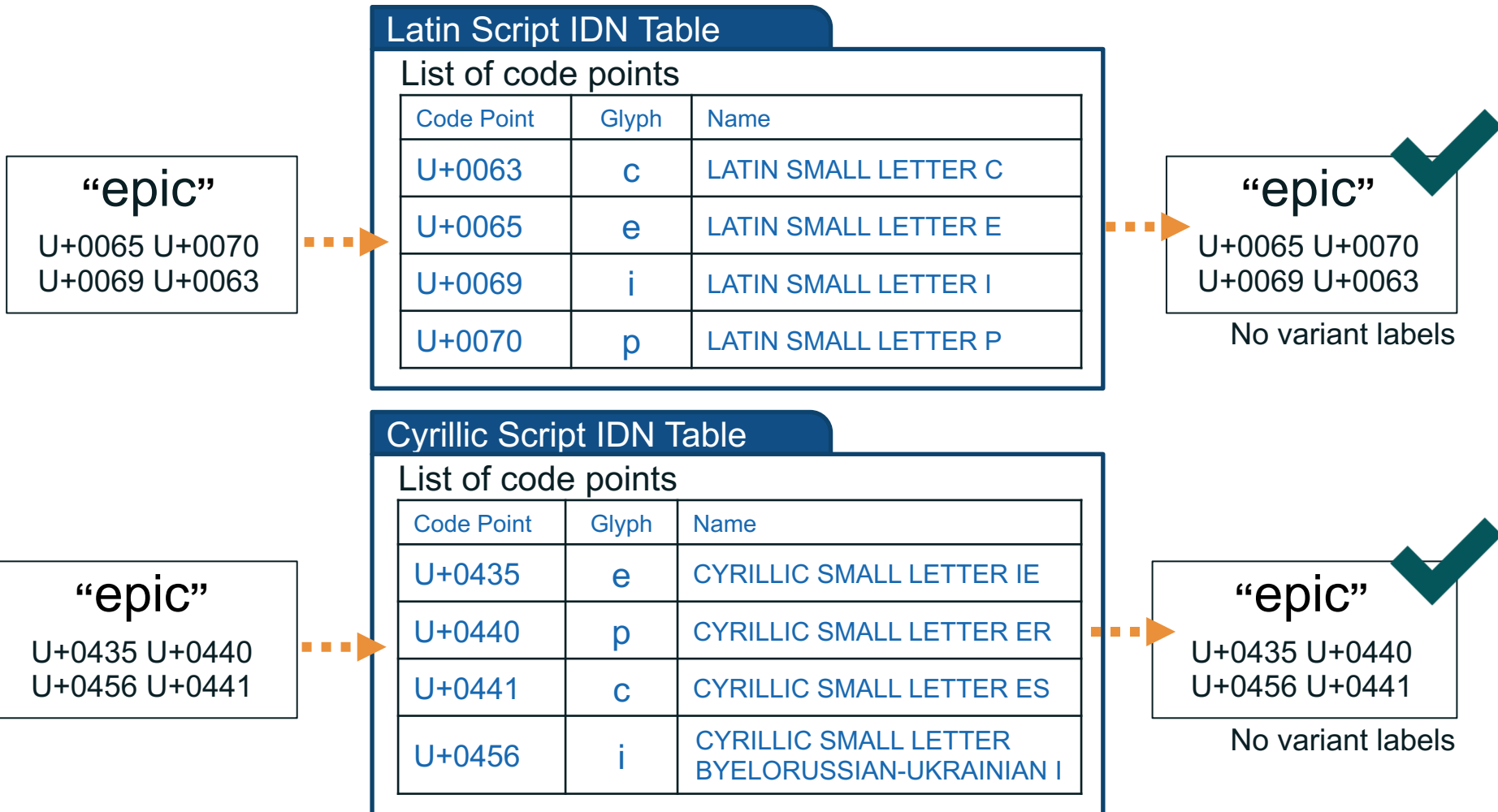
064A(ي) <-> 06CC(ى) blocked

## Result of harmonization

Now U+0628 U+064A U+0641 and U+0628 U+06CC U+0641 identified as blocked variants, addressing user confusion between “بيف.TLD” and “بيف.TLD”.

# Example: The Need of Harmonization for IDN Tables

- The two variant (visually identical) labels are created as non-variant labels under the two different IDN tables: **epic.TLD** and **epic.TLD**.



# Example: The Need of Harmonization for IDN Tables

- If a TLD offers both Latin and Cyrillic scripts, the IDN tables harmonized to identify variant labels across the IDN tables.

## Merged IDN Table

### List of code points

Code Point	Glyph	Name
U+0063	c	LATIN SMALL LETTER C
U+0065	e	LATIN SMALL LETTER E
U+0069	i	LATIN SMALL LETTER I
U+0070	p	LATIN SMALL LETTER P
U+0435	e	CYRILLIC SMALL LETTER IE
U+0440	p	CYRILLIC SMALL LETTER ER
U+0441	c	CYRILLIC SMALL LETTER ES
U+0456	i	CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I

Variant definition 0063(c) <-> 0441(c) blocked  
0065(e) <-> 0435(e) blocked  
0069(i) <-> 0456(i) blocked  
0070(p) <-> 0440(p) blocked

## Result of harmonization

Now U+0065 U+0070 U+0069  
U+0063 and U+0435 U+0440  
U+0456 U+0441 identified as  
blocked variants, addressing  
user confusion between  
“epic.TLD” and “epic.TLD”.



# Reference Label Generation Rules (LGRs)

- ◎ Reference LGRs based on RZ-LGR, and designed for the second level:
  - Formed in consultation with the relevant script community.
  - Follows IDNA2008 and design principles in [RFC6912](#) and SSAC [SAC060](#).
  - Developed in machine-readable [RFC 7940](#) format.
  - [Pre-vetted for security and stability considerations](#).
- ◎ Registry operators encouraged to consult the reference LGRs while they design their own IDN tables. Strict adherence not required.
- ◎ Reference LGRs used as baseline for reviewing IDN tables for gTLD registries.
- ◎ Currently, 46 reference LGRs are [published](#):
  - [30 language-based LGRs](#): Arabic, Belarusian, Bosnian (Cyrillic), Bosnian (Latin), Bulgarian, Chinese, Danish, English, Finnish, French, German, Hebrew, Hindi, Hungarian, Icelandic, Italian, Korean, Latvian, Lithuanian, Macedonian, Montenegrin, Norwegian, Polish, Portuguese, Russian, Serbian, Spanish, Swedish, Thai, and Ukrainian.
  - [16 script-based LGRs](#): Arabic, Bangla (Bengali), Devanagari, Ethiopic, Georgian, Gujarati, Gurmukhi, Hebrew, Kannada, Khmer, Lao, Malayalam, Oriya, Sinhala, Tamil, and Telugu.

# Background of IDN Table Review

---

- ⦿ A generic top-level domain (gTLD) registry operator (RO) intending to offer registrations in different languages and scripts.
  - Must be approved to offer IDN service for the languages and scripts.
  - Develop its IDN tables and share these with ICANN org for review.
  - Approved IDN tables updated in Exhibit A of the Registry Agreement.
  
- ⦿ ICANN reviews the IDN table(s) for security and stability, when an IDN table is added or updated by an RO through one of the following three processes.
  - Registry System Testing (RST) before delegation.
  - Registry Service Evaluation Policy (RSEP) process for IDN Service.
  - RST for registry service provider (RSP) change.
  
- ⦿ IDN tables are published at the [IANA Repository](#) to promote consistency.
  - The gTLD registry operators required to publish IDN tables.
  - The ccTLDs encouraged to publish their IDN tables.

# IDN Table Review Tool

---

- ⦿ Aims for developing the tool:
  - Increase efficiency in reviewing the IDN table.
  - Promote consistency of reviews.
  - Promote transparency of the reviews.
- ⦿ Available online for ICANN org as well as registry operators (ROs) and registry service providers (RSPs) for use, providing results efficiently.
  - Input: IDN table and relevant reference LGR.
  - Output: IDN table review report in HTML format.
- ⦿ Same review analysis available for ICANN org and contracted parties, making the process transparent.
- ⦿ Comparison always done with the published reference LGR, making the process consistent.

# Engage with ICANN – Thank You and Questions



One World, One Internet

Visit us at [icann.org](https://icann.org)



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



[facebook.com/icannorg](https://facebook.com/icannorg)



[youtube.com/icannnews](https://youtube.com/icannnews)



[flickr.com/icann](https://flickr.com/icann)



[linkedin/company/icann](https://linkedin/company/icann)



[soundcloud/icann](https://soundcloud/icann)



[instagram.com/icannorg](https://instagram.com/icannorg)