

LGR Toolset User Guide (Advanced Mode)



IDN Program
November 2023

Introduction to LGR Toolset (Advanced-mode)

- Label Generation Rulesets (LGRs) specify metadata, code point repertoire, variant rules and Whole Label Evaluation (WLE) rules to generate labels
- RFC 7940 describes how LGR can be specified using XML, a machine readable format
- LGR can be used to generate domain name labels for use in the internet's root zone and other levels
- LGR Toolset can be used to
 - create an LGR
 - view LGR as an HTML webpage or XML Format
 - merge multiple LGRs into a single LGR
 - validate single label or multiple labels using an LGR
 - determine cross-script variants of labels using a merged LGR
 - manage LGRs by comparing or combining them
 - review possible impact of a new or a revised LGR on existing labels
 - harmonize multiple LGRs
 - compute variants

Availability of LGR Toolset

- LGR Toolset is available with the following disclaimer:

THIS SOFTWARE IS PROVIDED BY ICANN AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ICANN OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- Online deployment
 - Visit <https://lgrtool.icann.org/>
- Open source package(s) release with BSD license
 - Released at github: [lgr-core](#), [lgr-django](#), [munidata](#), [picu](#)
- For queries or feedback
 - Email to IDNProgram@icann.org
- For further details, visit the [LGR Toolset webpage](#) or www.icann.org/idn

Agenda

1

Landing Page
(pp. 11-12)

2

Advanced LGR Tools
(pp. 13-14)

3

Import or Load LGR
(pp. 15-18)

4

Importing Multiple
LGRs
(pp. 19-22)

5

Create LGR
(pp. 23-49)

6

Validate LGR
(pp. 50-53)

Agenda (Cont'd)

7

Viewing LGR as XML
and HTML
(pp. 54-65)

8

Validating a Label
(pp. 66-71)

9

Review Impact on
Existing Labels by
Revising an Existing
LGR
(pp. 72-77)

10

Review Impact on
Existing Labels by
Introducing a New
LGR
(pp. 78-83)

11

Validating Multiple
Labels Using a Single
or Merged LGR(s)
(pp. 84-90)

12

Viewing Cross-Script
Variants of Labels
(pp. 91-96)

Agenda (Cont'd)

13

Compare LGRs
(pp. 97-101)

14

Harmonize
Multiple LGRs
(pp. 102-105)

15

Compute Variants
(pp. 106-110)

16

Label forms
(pp. 111-118)

Agenda Details

- **Landing Page**
- **Advanced LGR Tools**
- **Import or Load LGR**
 - Import LGR
 - Select LGR with validating repertoire
- **Import Multiple LGRs**
 - Import LGR
 - Select multiple LGRs with validating repertoire
 - Specify name of the merged LGR

Agenda Details (Cont'd)

- **Create LGR**

- Create LGR
- Enter LGR Details
- Add Code Points
- Expand Ranges
- Add References
- Define Metadata
- Define Classes, Rules & Actions
- Define Code Point Properties
- Download LGR

- **Validate LGR**

- **View LGR as XML and HTML**

- **Validate a Label**

- Validate a label using a single LGR
- Validate a label using a merged LGR

Agenda Details (Cont'd)

- **Review Impact on Existing Labels by Revising an Existing LGR**
 - Select “Diff labels of two LGRs”
 - Enter Details
 - Email Notification
 - Download Results

- **Review Impact on Existing Labels by Introducing a New LGR**
 - Select “Get collisions in a list of labels”
 - Enter Details
 - Email Notification
 - Download Results

- **Validate Multiple Labels Using a Single or Merged LGR(s)**
 - Select “Generate disp. annotations”
 - Enter Details
 - Email Notification
 - Download Results

Agenda Details (Cont'd)

- **View Cross-Script Variants of Labels**
 - Select “Cross-script variants”
 - Enter Details
 - Email Notification
 - Download Results
- **Compare LGRs**
 - Compare LGRs
 - Select Union, Intersection or Difference
 - With merged LGRs, “Diff” chosen by default
- **Harmonize Multiple LGRs**
 - Select “Harmonization”
 - Enter Details
 - Email Notification
 - Download Results
- **Compute Variants**
 - Select “Compute Variants”
 - Select relevant LGR
 - Upload list of labels and enter email
 - Download results on email notification
- **Label Forms**

Landing Page

Landing Page of the LGR tool

- A user guide with more information is published on the [LGR Tool webpage](#).
- Go to link - Use the **Label Generation Rules Tool**

🏠 LGR Tools

Tasks

Help ▾

🔍 Label forms

👤 lgr-user ▾

Welcome to the LGR (Label Generation Ruleset) Tools

Select your mode:

Basic Mode

✓ Validate label(s) against an LGR

🔍 Review IDN table(s)

🔧 Advanced LGR Tools

Advanced Mode

IDN Table Reviewing

Advanced LGR Tools


Advanced LGR (Label Generation Ruleset) Tools

This application provides a convenient interface for browsing and editing [LGRs](#) conforming to the [Representing Label Generation Rulesets using XML](#) specification.

Previously loaded LGR file(s)

No LGR has been previously loaded.

Create a new LGR file or import an existing one

 Import an existing XML file

⚠ Note that importing large LGR files may take significant time to load on your browser.

 Start with a New blank XML file

Start from a built-in LGR

The following LGRs are pre-installed in the system. You may use them as a starting point for your own LGR. To do so, just click on it to make a copy that you can then edit.

- [Open Sample-French](#)
- [Open RZ-LGR 1](#)
- [Open RZ-LGR 2](#)
- [Open RZ-LGR 3](#)
- [Open RZ-LGR 4](#)
- [Open RZ-LGR 5](#)


Remember to save your work regularly by downloading a copy of the XML file.

Please send any feedback to support@viagenie.ca.

Your saved results

The following files contains your tools computation results.

⚠ Note that these files could be cleaned up regularly.

- [Download 20221022_171249_annotation_RZ-LGR_5.txt.gz](#) 

Import or Load LGR

Import or Load LGR

🏠 LGR Tools

📁 Import

📄 New

⚙️ Tools ▾

[Switch mode](#)

🔍 Label

Advanced LGR (Label Generation Ruleset) Tools

This application provides a convenient interface for browsing, editing and [importing LGRs](#) conforming to the [Representing Label Generation Rulesets using XML](#) specification.

Previously loaded LGR file(s)

No LGR has been previously loaded.

To start by using an existing LGR file in XML format, click on the “Import” button

Create a new LGR file or import an existing one

📁 Import an existing XML file

⚠️ **Note that importing large LGR files may take significant time to load on your browser.**

📄 Start with a New blank XML file

Select LGR with Validating Repertoire

Import existing LGR

To import LGR set, select the XML files composing the LGR set. The

ⓘ Import function will only perform some basic checking (example: XML validity). Once the LGR has been imported, you need to use the Validate LGR button to do a complete check.

Select file(s)

Choose Files No file chosen

If you select more than one file, this will create a LGR set. File must be encoded in UTF-8 and using 0x0A line ending.

Validating repertoire

RZ-LGR 5
idna2008_10.0.0
msr-4-wle-rules-25jan19-en

⚠ Note that importing large LGR files may take significant time to load on your browser.

1. To import or load an existing LGR in XML format, click on “Choose Files”

2. Choose the “Validating repertoire” from the given options

3. Click on the “Import” button

Imported LGR

🏠 LGR Tool / proposal-lgr-malayalam-20180627 - Proposed LGR for Malayalam

📁 Import 📄 New ⚙️ Tools ▾ Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🏷️ Label forms ★ About

Code points **References** Meta data Tags Rules

Add code point(s)

➔ Select code point(s) to apply batch action... Show **500** entries

Search:

<input type="checkbox"/>	Code point ▲	Character Name ⇅	Tags ⇅	Comments ⇅	Action
<input type="checkbox"/>	U+0B02 (ꣲ) 2 Variant(s)	ORIYA SIGN ANUSVARA		not part of repertoire	See code point
<input type="checkbox"/>	U+0B03 (ꣳ) 2 Variant(s)	ORIYA SIGN VISARGA		not part of repertoire	See code point
<input type="checkbox"/>	U+0B20 (꣠) 2 Variant(s)	ORIYA LETTER TTHA		not part of repertoire	See code point
<input type="checkbox"/>	U+0B9C (꣚) 2 Variant(s)	TAMIL LETTER JA		not part of repertoire	See code point
<input type="checkbox"/>	U+0BAE (꣞) 2 Variant(s)	TAMIL LETTER MA		not part of repertoire	See code point
<input type="checkbox"/>	U+0BB5 (꣥) 2 Variant(s)	TAMIL LETTER VA		not part of repertoire	See code point
<input type="checkbox"/>	U+0BBF (꣟) 2 Variant(s)	TAMIL VOWEL SIGN I		not part of repertoire	See code point
<input type="checkbox"/>	U+0BC6 (꣦) 2 Variant(s)	TAMIL VOWEL SIGN E		not part of repertoire	See code point
<input type="checkbox"/>	U+0BC7 (꣧) 2 Variant(s)	TAMIL VOWEL SIGN EE		not part of repertoire	See code point

The screen looks like this after successful import of existing LGR file in XML format

Import Multiple LGRs

Import or Load LGRs

🏠 LGR Tools

📁 Import

📄 New

⚙️ Tools ▾

[Switch mode](#)

🏷️ Label

Advanced LGR (Label Generation Ruleset) Tools

This application provides a convenient interface for browsing and managing LGRs configurations and L specifications.

Previously loaded LGR file(s)

No LGR has been previously loaded.

Create a new LGR file or import an existing one

📁 Import an existing XML file

⚠️ Note that importing large LGR files may take significant time to load on your browser.

📄 Start with a New blank XML file

To start by using existing LGR files in XML format, click on the “Import” button

Select LGRs with Validating Repertoire

🏠 LGR Tool

📁 Import

📄 New

⚙️ Tools ▾

[Switch mode](#)

🔍 Label forms

★ About

Import existing LGR

To import LGR set, select the XML files composing the LGR set.

ⓘ Import function will only perform some basic checking. For a complete check, you need to use the Validate LGR button to do a complete check.

1. To import or load multiple LGRs in XML format, click on “Choose Files” and select multiple files

Select file(s)

Choose Files 4 files

If you select more than one file, this will create a LGR set

Validating repertoire

msr-3-wle-rules-28mar18-en

Code points will be limited to the selected repertoire

LGR set name

multiple-lgrs

The name of the set

2. Choose the “Validating repertoire” from the given options

Import

⚠️ Note that importing large LGR files may take a significant time

3. Enter a name for the set of LGRs

4. Click on the “Import” button

English (en) ▾ Go

Imported LGRs As a Merged LGR

🏠 LGR Tool / multiple-lgrs - Proposed LGR for Arabic Script|Proposed LGR for Armenian|...

📁 Import 📄 New ⚙️ Tools ▾ [Switch mode](#) ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🏷️ Label forms ★ About

Code points [References](#) [Meta data](#) [Tags](#) [Rules](#) [Embedded LGRs](#)

Show entries

Search:

Code point ▲	Character Name ⇅	Tags ⇅	Comments ⇅	Action
U+0067 (g) 2 Variant(s)	LATIN SMALL LETTER G	Latin	Cross-script homoglyph	See code point
U+0068 (h) 3 Variant(s)	LATIN SMALL LETTER H	Latin	Cross-script homoglyph	See code point
U+006E (n) 2 Variant(s)	LATIN SMALL LETTER N	Latin	Cross-script homoglyph	See code point
U+006F (o) 4 Variant(s)	LATIN SMALL LETTER O		Cross-script homoglyph	See code point
U+0071 (q) 2 Variant(s)	LATIN SMALL LETTER Q		Cross-script homoglyph	See code point
U+0075 (u) 3 Variant(s)	LATIN SMALL LETTER U		Cross-script homoglyph	See code point
U+0269 (i) 3 Variant(s)	LATIN SMALL LETTER IOTA	Latin	Cross-script homoglyph	See code point
U+03B7 (η) 2 Variant(s)	GREEK SMALL LETTER ETA	Greek	Cross-script homoglyph	See code point
U+03B9 (ι) 3 Variant(s)	GREEK SMALL LETTER IOTA	Greek	Cross-script homoglyph	See code point

The screen looks like this after successful import of multiple LGR files in XML format

Create LGR

Create an LGR

🏠 LGR Tools

📁 Import

📄 New

⚙️ Tools ▾

[Switch mode](#)

🔍 Label

Advanced LGR (Label Generation Ruleset) Tools

This application provides a convenient interface for browsing and editing Label Generation Rulesets using XML specification.

Previously loaded LGR file(s)

No LGR has been previously loaded.

Create a new LGR file or import an existing one

📁 Import an existing XML file

⚠️ Note that importing large LGR files may take significant time to load on your browser.

📄 Start with a New blank XML file

To start by creating an LGR file in XML format, click on the “New” button

Enter LGR Details

Switch mode

1. Write name for the LGR being created

Create a new LGR

Name

Validating repertoire

RZ-LGR 5
idna2008_10.0.0
msr-4-wle-rules-25jan19-en

2. Select Validating repertoire from the given options for the “New” LGR. MSR for RZ-LGR and IDNA version for second level LGRs

3. Click on the “Create” button

Add Code Points

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools

[Switch mode](#) ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🗒 Label forms ★ About

Code points [References](#) [Meta data](#) [Tags](#) [Rules](#)

→ Select code point(s) to apply batch action... ▾ Show entries Search:

<input checked="" type="checkbox"/>	Code point	Character Name	Tags	Comments	Action
No data available in table					

Previous Next

English (en) ▾ Go

©

Click on “Add code points” to add code points to the newly created LGR

Add Code Points

To add code points, there are four different ways

Add code point(s)

Code point

Code point range

Code point from script

Import from file

Code point

Override repertoire

Add Code Point

1. Add code points one by one

2. Add code points by giving a range

3. Add code points by specifying a script

4. Add code points from a file

Add Code Points

The screenshot shows the 'LGR Tool / new-lgr' interface. A modal dialog titled 'Add code point(s)' is open, featuring four tabs: 'Code point', 'Code point range', 'Code point from script', and 'Import from file'. The 'Code point' tab is active, showing a text input field with '1780' and an 'Add Code Point' button. An 'Override repertoire' checkbox is also present. Two orange callout boxes provide instructions: the first points to the input field, and the second points to the 'Add Code Point' button.

1. Write the code point to be added. The code point value or the actual character can be entered

2. Click on the “Add Code Point” button

Add Code Points

The screenshot shows the 'Add code point(s)' dialog in the LGR Tool. The dialog has four tabs: 'Code point', 'Code point range', 'Code point from script', and 'Import from file'. The 'Code point range' tab is selected. It contains two input fields: 'First code point' with the value '1781' and 'Last code point' with the value '179A'. A blue 'Next' button is located at the bottom right of the dialog. Three orange callout boxes provide instructions: the first points to the '1781' field, the second points to the '179A' field, and the third points to the 'Next' button. The background shows the LGR Tool interface with 'Code points', 'References', and 'Meta' tabs, and a 'Go' button at the bottom left.

1. Add the first code point of the range

2. Add the last code point of the range

3. Click on "Next" button

Either code point value or the character can be added

Add Code Points

The screenshot shows the 'Add code point(s)' dialog box with the following elements:

- Code point** (selected), **Code point range**, **Code point from script**
- Script**: Tamil
- Validating repertoire**: RZ-LGR 5 (selected), idna2008_10.0.0, msr-4-wle-rules-25jan19-en
- Manual import**
- Next** button

Four numbered callouts provide instructions:

1. Select a script from the given options
2. Select Validating Repertoire from the given options
3. "Manual import" is optional – allows checking each code point in the script before adding to the LGR
4. Click on "Next" button

Add Code Points

The screenshot shows the 'Add code point(s)' dialog box in the LGR Tool. The dialog has four tabs: 'Code point', 'Code point range', 'Code point from script', and 'Import from file'. The 'Import from file' tab is selected. It contains a 'Select a file' section with a 'Choose File' button and the filename 'Code Points.txt'. Below this is a note: 'File containing data to be imported. File must be encoded in UTF-8 and using UNIX line ending.' The 'File type' section has a dropdown menu set to 'RFC3743' and an unchecked checkbox for 'Manual import'. A blue 'Next' button is at the bottom right. Four orange callout boxes provide instructions: 1. '1. Select the file containing code points' points to the file selection area. 2. '2. Select the file type from the given options' points to the 'File type' dropdown. 3. '3. "Manual import" is optional – allows checking each code point in the file before adding to the LGR' points to the 'Manual import' checkbox. 4. '4. Click on the "Next" button' points to the 'Next' button.

1. Select the file containing code points

2. Select the file type from the given options

3. "Manual import" is optional – allows checking each code point in the file before adding to the LGR

4. Click on the "Next" button

Expand Ranges

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🔍 Label forms ★ About

Code points References Meta data Tags Rules

➔ Select code point(s) to apply batch action... ▾

Search:

<input type="checkbox"/>	Code point		Comments	Action
<input type="checkbox"/>	U+1780 (ក) 0 Variant(s)		KHMER LETTER KA	See code point
<input type="checkbox"/>	U+1781 (ខ) ... U+1784 (ឃ)		KHMER LETTER KHA ... KHMER LETTER NGO	See code point Expand range
<input type="checkbox"/>	U+1787 (ង) 0 Variant(s)		KHMER LETTER CO	See code point
<input type="checkbox"/>	U+1788 (ច) 0 Variant(s)		KHMER LETTER CHO	See code point
<input type="checkbox"/>	U+1789 (ឈ) 0 Variant(s)		KHMER LETTER NYO	See code point
<input type="checkbox"/>	U+178A (ត) 0 Variant(s)		KHMER LETTER DA	See code point
<input type="checkbox"/>	U+178B (ថ) 0 Variant(s)		KHMER LETTER TTHA	See code point
<input type="checkbox"/>	U+178D (ឍ) 0 Variant(s)		KHMER LETTER TTHO	See code point

To expand all the code points in all the ranges of the LGR, click on “Expand range(s)”

To expand all the code points in this range, click on “Expand range”

Add References

Click on "References" tab to add references to the LGR

LGR Tool / new-lgr

Import New Tools

Switch mode Validate label Validate LGR Output Label forms About

Code points **References** Meta data Tags Rules

Existing references

Reference id	Description	URL	Action
			Save

1. Add Reference id 2. Add comments 3. Add detailed reference

New reference

Reference id	Description	URL
<input type="text" value="101"/>	<input type="text" value="A reference Grammar of Punjabi"/>	<input type="text" value="http://pt.learnpunjabi.org/assets/A%2C"/>

[Add](#)

4. Click on the "Add" button

English (en) Go

Add References

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🏷️ Label forms ★ About

New reference created ✕

Code points **References** Meta data Tags Rules

Existing references

Reference id	Description	URL	Action
101	A reference Grammar of Punjabi	http://pt.learnpunjabi.org/assets/A%20Reference%20Gramm	🗑️

Save

New reference

Reference id	Description	URL
101	A reference Grammar of Punjabi	http://pt.learnpunjabi.org/assets/A%2C

Add

English (en) ▾ Go

©

“Existing references” tab shows the added references

Define Metadata

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾

[Switch mode](#) ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 📄 Label forms ★ About

[Code points](#) [References](#) **Meta data** [Tags](#) [Rules](#)

Version

Version comment

Date

Language

Scope

Scope type

Validity start

Validity end

Unicode version

Description

Click on "Metadata" tab to add meta information about the LGR

Define Meta Data

Second half of the
“Metadata” tab

LGR Tool / new-lgr

Import New Tools

Switch mode Validate label Validate LGR Output Label forms About

Scope type domain

Validity start

Validity end

Unicode version 10.0.0

Description

<h1>Label Generation Rules for Khmer</h1>
<h2>Overview</h2>
<p>For more details on this proposal see "Proposal for a Khmer Script Root Zone LGR [Proposal]" </p>
<h2>Repertoire</h2>
<p>According to Section 5 "Repertoire", in "[Proposal]".</p>
<h2>Variants</h2>
<p>According to Section 6 "Variants", in "[Proposal]", two Khmer characters are captured by a variant relation between the two subscript sequences.
<h2>Character Classes</h2>
<p>Some consonants have been given the following character classes:

Description type text/html

Validating repertoire msr-3-wle-rules-28mar18-en

Cancel Save

English (en) Go

Add any text for
describing the LGR

Select “Description
type” from the given
options

Select “Validating repertoire”
from the given options

After filling every detail,
click on the “Save” button



View Tags

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾ [Switch mode](#) ✓ Validate label 📄 Validate LGR ↻ Output ▾ 🏷️ Label forms ★ About

[Code points](#) [References](#) [Meta data](#) **Tags** [Rules](#)

Existing tags

Tag name	Associated code points	Action
vowel	U+17A7 (ḡ)	
consonant	U+1781 (ḡ) U+1782 (ḡ) U+1783 (ḡ) U+1784 (ḡ) U+1787 (ḡ)	

i To create a new tag: Add it to a code point first to find it back in this list

English (en) ▾ Go

©

Click on "Tags" tab to view all tag names and associated code points

Define Classes, Rules & Actions

The screenshot shows the LGR Tool interface for a new LGR. The top navigation bar includes 'Import', 'New', and 'Tools' buttons, along with a 'Switch mode' link and utility buttons for 'Validate label', 'Validate LGR', 'Output', 'Label forms', and 'About'. Below this is a tabbed interface with 'Code points', 'References', 'Meta data', 'Tags', and 'Rules' tabs. The 'Rules' tab is active, showing three sections: 'Classes', 'Rules', and 'Actions'. Each section has a '+ New class', '+ New rule', and '+ New action' button respectively. Three orange callout boxes provide instructions: 'Click on "Rules" tab to add certain rules to the LGR' (pointing to the 'Rules' tab), 'Click on "New class" button to add classes to the LGR' (pointing to the '+ New class' button), 'Click on "New rule" button to add rules to the LGR' (pointing to the '+ New rule' button), and 'Click on "New action" button to add actions to the LGR' (pointing to the '+ New action' button'). At the bottom left, there is a language selector set to 'English (en)' and a 'Go' button. A small copyright symbol is visible at the bottom right of the interface.

Click on "Rules" tab to add certain rules to the LGR

Click on "New class" button to add classes to the LGR

Click on "New rule" button to add rules to the LGR

Click on "New action" button to add actions to the LGR

Define Classes

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR ↻ Output ▾ 🗒 Label forms ★ About

Code points References Meta data Tags **Rules**

Classes

```
<class name="consonant" from-tag="consonant" comment="Any consonant" />
```

💾 Save ✕ Cancel

Rules

Actions

English (en) ⌵ Go

©

1. Add classes in the relevant box

2. Click on the "Save" button

Define Rules

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR ↻ Output ▾ 🗒 Label forms ★ About

Classes

```
<class name="consonant" from-tag="consonant" comment="Any consonant" />
```

Rules

📁 Save ✕ Cancel

```
<rule name="follows-consonant" comment="WLE Rule No. 8: checks if sign code point or subscript consonant follows a consonant">  
  <look-behind>  
    <class by-ref="consonant" />  
  </look-behind>  
  <anchor />  
</rule>
```

1. Add the rule in the relevant box

2. Click on the "Save" button

Define Actions

↑ LGR Tool / new-lgr

Import

New

Tools ▾

Switch mode

✓ Validate label

Validate LGR

Output ▾

Label forms

★ About

Rules

```
<rule name="follows-consonant" comment="WLE Rule No. 8: checks if sign code point or subscript consonant follows a  
consonant">  
  <look-behind>  
    <class by-ref="consonant" />  
  </look-behind>  
  <anchor />  
</rule>
```

1. Add action in the relevant box

2. Click on the "Save" button

Actions

```
<action disp="blocked" any-variant="blocked" />
```

Save Cancel

English (en) ▾ Go

©

Apply Batch Action

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🏷️ Label forms ★ About

Code points References Meta data Tags Rules

Expand range(s) Add code point(s)

Search:

✓ → Apply batch action...
Add WLE
Add Tags

Show 500

2. Select the type of batch action

1. Select multiple code points to apply the batch action

		Character Name	Tags	Comments	Action
<input type="checkbox"/>	U+1780 (ក) 1 Variant(s)	KHMER LETTER KA			See code point
<input type="checkbox"/>	U+1781 (ខ) ... U+1784 (គ)	KHMER LETTER KHA ...			See code point Expand range
<input type="checkbox"/>	U+1785 (ឃ) 2 Variant(s)	KHMER		ded from out-of-repertoire variant	See code point
<input type="checkbox"/>	U+1787 (ង) 0 Variant(s)	KHMER LETTER CO	consonant		See code point
<input checked="" type="checkbox"/>	U+1788 (ច) 0 Variant(s)	KHMER LETTER CHO			See code point
<input checked="" type="checkbox"/>	U+1789 (ឆ) 0 Variant(s)	KHMER LETTER NYO			See code point
<input checked="" type="checkbox"/>	U+178A (ជ) 0 Variant(s)	KHMER LETTER DA			See code point
<input checked="" type="checkbox"/>	U+178B (ត) 0 Variant(s)	KHMER LETTER TTHA			See code point

Apply Batch Action – Add WLE

The screenshot shows the 'LGR Tool / new-lgr' interface. A modal dialog titled 'Add Rule' is open. The 'when' dropdown is set to 'follows-consonant'. The 'not-when' dropdown is empty. Below the dropdowns, it says 'This rule will be added to 4 codepoint(s)'. A blue 'Next' button is at the bottom right of the dialog. Two orange callout boxes are overlaid on the dialog:

1. Add when-rule / not-when-rule from the list of WLE rules
2. Click on "Next" button

In the background, a table of code points is visible:

Code point	Character	Variant(s)	Category
U+1780	𑄀	1 Variant(s)	KHMER LETTER NYO
U+1781	𑄁	U+1784	KHMER LETTER NYO
U+1785	𑄂	2 Variant(s)	KHMER LETTER NYO
U+1787	𑄄	0 Variant(s)	KHMER LETTER NYO
U+1788	𑄅	0 Variant(s)	KHMER LETTER NYO
U+1789	𑄆	0 Variant(s)	KHMER LETTER NYO
U+178A	𑄇	0 Variant(s)	KHMER LETTER DA

Apply Batch Action – Add Tags

The screenshot shows the LGR Tool interface with the 'Add Rule' dialog box open. The dialog has a 'Tags' input field containing 'consonant' and 'vowel', with 'vowel' selected. A 'Next' button is visible. Two orange callout boxes provide instructions: '1. Type new tags, separate by a space, or select the existing ones' and '2. Click on "Next" button'.

1. Type new tags, separate by a space, or select the existing ones

2. Click on "Next" button

Define Code Point Properties

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR ↻ Output ▾ 📄 Label forms ★ About

Code points References Meta data Tags Rules

Expand range(s) Add code point(s)

⇒ Select code point(s) to apply batch action... Show 500 entries Search:

<input type="checkbox"/>	Code point		Tags	Comments	Action
<input type="checkbox"/>	U+1780 (𑄀) 0 Variant(s)				See code point
<input type="checkbox"/>	U+1781 (𑄁) ... U+1784 (𑄄)	KHMER LETTER KHA ... KHMER LETTER NGO	consonant		See code point Expand range
<input type="checkbox"/>	U+1787 (𑄇) 0 Variant(s)	KHMER LETTER CO	consonant		See code point
<input type="checkbox"/>	U+1788 (𑄈) 0 Variant(s)	KHMER LETTER CHO			See code point
<input type="checkbox"/>	U+1789 (𑄉) 0 Variant(s)	KHMER LETTER NYO			See code point
<input type="checkbox"/>	U+178A (𑄊) 0 Variant(s)	KHMER LETTER DA			See code point
<input type="checkbox"/>	U+178B (𑄋) 0 Variant(s)	KHMER LETTER TTHA			See code point

Click on the “See code point” button to add code point details/properties

Define Code Point Properties - 1

🏠 LGR Tool / new-lgr / U+1780 (ក) KHMER LETTER KA

📁 Import 📄 New ⚙️ Tools ▾

📄 Validate LGR 🔄 Output ▾ 📄 Label forms ★ About

Code points / U+1780 (ក) KHMER LETTER KA

Code point appeared in Unicode version: 3.0.0

Variants

Code point: Override repertoire: **Add variant**

Code point	Type	Comments	When	Not When	References	Action
U+1785 (ក) KHMER LETTER KA Age: 3.0.0.0	<input type="text" value="blocked"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		Delete variant Edit references

Properties

Tags
 space-separated tags

When

Not-When

Comment

1. Add variant for the code point

2. Click on the “Add variant button”

3. Add details for the added variant – type, comments, when-rule and not-when rule

4. Add tags for the code point

5. Add when-rule/not-when rule from the list of rules it provides

6. Add any description for the code point

Define Code Point Properties - 2

🏠 LGR Tool / new-lgr / U+1780 (ក) KHMER LETTER KA

📁 Import 📄 New ⚙️ Tools

Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output 📄 Label forms ★ About

Tags

✖ consonant

space-separated tags

When

follows-consonant

Not-When

Comment

Any text description for the code point

Save variants, tags, context rules and comment

7. Click on “Save variants, tags, context rules and comment” button

Note: If the added code point is not in the repertoire, the system will automatically add the out-of-repertoire mapping.

References

No references associated with code point.

Edit

8. Click on the “Edit” button to add references to the code point

Delete code point

To delete code point & its details, click on the “Delete code point” button

English (en) Go

Define Code Point Properties - 3

🏠 LGR Tool / lgr-2-khmer-script-26jul17-en - Root Zone LGR

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🗉 Label forms ★ About

Code points References Meta data Tags Rules

Populate variants Add code point(s)

➡ Select code point(s) to apply batch action... ⌵ Show 500 ⌵ entries Search:

<input type="checkbox"/>	Code point ▲	Character Name ⌵	Tags ⌵	Comments ⌵	Action
<input type="checkbox"/>	U+0E01 (n) 3 Variant(s)	THAI CHARACTER KO KAI		Automatically added to repertoire variant	See code point
<input type="checkbox"/>	U+0E16 (n) 2 Variant(s)	THAI CHARACTER THO THUNG		Automatically added to repertoire variant	See code point
<input type="checkbox"/>	U+1780 (n) 1 Variant(s)	KHMER LETTER KA	consonant,sc:Khmr,series-three	Khmer	See code point

The screen looks like this after successfully defining code point properties.

If the variant definitions are not symmetric, this button will show up. Click “Populate variants” to automatically populate variant mappings.

Download LGR

🏠 LGR Tool / new-lgr

📁 Import 📄 New ⚙️ Tools

Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output 🔍 Label forms ★ About

Code points References Meta data Tags Rules

➔ Select code point(s) Search:

👁 View XML
⬇ Download
🌐 HTML Output

range(s) Add code point(s)

Click on "Output" and select "Download" from the given options to download the created LGR.

<input type="checkbox"/>	Code point				Action
<input type="checkbox"/>	U+1780 (័) 1 Variant(s)				See code point
<input type="checkbox"/>	U+1781 (៑) ... U+1784 (៌)	KHMER LETTER KHA ... KHMER LETTER NGO	consonant		See code point Expand range
<input type="checkbox"/>	U+1785 (៍) 2 Variant(s)	KHMER LETTER CA		Automatically added from out-of-repertoire variant	See code point
<input type="checkbox"/>	U+1787 (៏) 0 Variant(s)	KHMER LETTER CO	consonant		See code point
<input type="checkbox"/>	U+1788 (៎) 0 Variant(s)	KHMER LETTER CHO			See code point
<input type="checkbox"/>	U+1789 (៏) 0 Variant(s)	KHMER LETTER NYO			See code point
<input type="checkbox"/>	U+178A (័) 0 Variant(s)	KHMER LETTER DA			See code point

Validate LGR

Validate LGR

🏠 LGR Tool / proposed-arabic-lgr-18092017-en - Proposed LGR for Arabic Script

📁 Import 📄 New ⚙️ Tools ▾ [Switch mode](#) ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🏷️ Label forms ★ About

Code points References Meta data Tags Rules

➡ Select code point(s) to apply batch action... ⚙️ Show 500 entries Search:

Expand range(s) Add code point(s)

<input type="checkbox"/>	Code point ▲	Character Name	Tags ⚙️	Comments ⚙️	Action
<input type="checkbox"/>	U+0620 () 0 Variant(s)	ARABIC	b		See code point
<input type="checkbox"/>	U+0621 (◌) 0 Variant(s)	ARABIC	b		See code point
<input type="checkbox"/>	U+0622 (◌) 4 Variant(s)	ARABIC	b		See code point
<input type="checkbox"/>	U+0623 (◌) 4 Variant(s)	ARABIC	b		See code point
<input type="checkbox"/>	U+0624 (◌) 1 Variant(s)	ARABIC	ab		See code point
<input type="checkbox"/>	U+0625 (◌) 4 Variant(s)	ARABIC LETTER ALEF WITH HAMZA BELOW	sc:Arab		See code point
<input type="checkbox"/>	U+0626 (◌) 7 Variant(s)	ARABIC LETTER YEH WITH HAMZA ABOVE	sc:Arab		See code point
<input type="checkbox"/>	U+0627 (◌)				See code point

Click on “Validate LGR” button to validate LGR and get a summary of the entire LGR. It is important to note that this function should be used for checking an LGR before use, e.g. when it is created or imported

Validate LGR

LGR Tool / proposed-arabic-lgr-18092017-en - Proposed LGR for Arabic Script

Validate LGR

Testing XML validity using RNG
XML is valid and complies with RNG.

Testing symmetry
Symmetry OK

Testing transitivity
Transitivity OK

Testing conditional variants
Conditional variants OK

Rebuilding LGR with Unicode version 6.3.0 and validating repertoire 'msr-3-wle-rules-28mar18-en'
LGR rebuild OK

Generate stats
General stats

- Number of code points: 113

↓ Save results Close

Code points

→ Select code po

<input type="checkbox"/>	Code p			
<input type="checkbox"/>	U+0620 (0 Variant(s)		
<input type="checkbox"/>	U+0621 (0 Variant(s)		
<input type="checkbox"/>	U+0622 (4 Variant(s)		
<input type="checkbox"/>	U+0623 (4 Variant(s)		
<input type="checkbox"/>	U+0624 (1 Variant(s)		
<input type="checkbox"/>	U+0625 (4 Variant(s)		
<input type="checkbox"/>	U+0626 (s)	7 Variant(s)	ARABIC LETTER YEH WITH HAMZA ABOVE	sc:Arab
<input type="checkbox"/>	U+0627 (!)	4 Variant(s)	ARABIC LETTER ALEF	sc:Arab
<input type="checkbox"/>	U+0628 (→)	0 Variant(s)	ARABIC LETTER BEH	sc:Arab

Summary of checks performed, including symmetry and transitivity

View LGR as XML and HTML

View LGR As XML

🏠 LGR Tool / proposal-thai-lgr-25may17-en - Thai Script Root Zone LGR Version 6.9

📁 Import 📄 New ⚙️ Tools ▾ [Switch mode](#) ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🏷️ Label forms ★ About

Code points References Meta data Tags Rules

➡ Select code point(s) to apply batch actions Search:

Click on “Output” and select “View XML” from the given options to get an XML view of the LGR

<input type="checkbox"/>	Code point	Tags	Comments	Action
<input type="checkbox"/>	U+0E01 (๑) 0 Variant(s)	cons		See code point
<input type="checkbox"/>	U+0E02 (๒) 0 Variant(s)	THAI CHARACTER KHO KHAU	cons	See code point
<input type="checkbox"/>	U+0E03 (๓) 0 Variant(s)	THAI CHARACTER KHO KHUAT	cons	See code point
<input type="checkbox"/>	U+0E04 (๔) 0 Variant(s)	THAI CHARACTER KHO KHWAI	cons	See code point
<input type="checkbox"/>	U+0E05 (๕) 0 Variant(s)	THAI CHARACTER KHO KHON	cons	See code point
<input type="checkbox"/>	U+0E06 (๖) 0 Variant(s)	THAI CHARACTER KHO RAKHANG	cons	See code point
<input type="checkbox"/>	U+0E07 (๗) 0 Variant(s)	THAI CHARACTER NGO NGU	cons	See code point
<input type="checkbox"/>	U+0E08 (๘) 0 Variant(s)	THAI CHARACTER CHO CHAN	cons	See code point
<input type="checkbox"/>	U+0E09 (๙) 0 Variant(s)	THAI CHARACTER CHO CHING	cons	See code point
<input type="checkbox"/>	U+0E0A (๐) 0 Variant(s)	THAI CHARACTER CHO CHANG	cons	See code point

View LGR As XML

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<lgr xmlns="urn:ietf:params:xml:ns:lgr-1.0">
  ▼<meta>
    <version comment="Thai Script Root Zone LGR Version 6.9">2</version>
    <date>2017-05-25</date>
    <unicode-version>6.3.0</unicode-version>
    <language>und-Thai</language>
    <scope type="domain">.</scope>
  ▼<description type="text/html">
    ▼<![CDATA[
      <h1>Label Generation Rules for the Thai Script</h1> <h2>Overview</h2> <p>This file contains Label
      Generation Rules (LGR) for the Thai script as would be appropriate for the Root zone. For more details on
      this LGR see "Proposal for a Thai Script Root Zone LGR [Proposal]" </p> <h2>Repertoire</h
    ]]>
    ▼<![CDATA[
      2> <p>In addition to the 68 code points according to Section 5 "Repertoire" in [Proposal], three sequences
      have been defined. The sequence U+0E4D U+0E32 was defined to replace the disallowed U+0E33 (THAI CHARACTER
      SARA AM) and to facilitate implementation of WLE rule <b>follows-consonant-tone</b> as a context rule. The
      other two sequences were defined to restrict U+0E45 (THAI CHARACTER LAKKHANGYAO) from appearing in any
      context other than these sequences. Accordingly, while U+0E45 is not listed by itself it brings the total
      of distinct code points to 69.</p> <h2>Variants</h2> <p>According to Section 6 "Variants", in "[Proposal]",
      this LGR defines no variants.</p> <h2>Character Classes</h2> <p>The Thai Script is an abugida in which
      consonant-vowel sequences are written as a unit: each unit is based on a consonant letter, and vowel, tone
      mark or diacritic notation are secondary. It is written with the combining marks stacked above or below the
      base consonant, like diacritics in European languages. However, although the concepts are quite similar,
      the implementations are significantly different.</p> <p>There are 44 characters that are classified as
      consonants, code points from this subset have been given the tag "cons". </p> <p>The 18 vowel symbols
      pronounced after a consonant are non-sequential: they can be located before (lv) , after (fv), above (av)
      or below (bv) the consonant, or in a combination of these positions, code points from this subset have been
      given the tag "fv1", "fv2", "fv3", "av", "bv", "lv". There are three code point sequences defined that include
      vowels. (Code point sequences do not carry tag values; instead, for code point sequences the subset values
      are identified in comments).</p> <p>There are 5 phonemic tones: mid, low, falling, high, and rising. These
      5 tones are represented by 4 tone marks plus the absence of a mark. Code points from this subset have been
      given the tag "tone"</p> <p>There are 3 diacritic symbols that have been included here and given the tag
      "ad". They differ in their frequency and purpose of usage. See also the discussion in section 5.4 in
      [Proposal].</p> <ul> <li>U+0E47 (MAITAIKHU) and U+0E4C (THANTHAKHAT) are commonly used in everyday
      communicating words</li> <li>U+0E4D (NIKHAHIT) is included because of its use to decompose U+0E33 (SARA AM,
```

XML view of the LGR

View LGR As HTML

🏠 LGR Tool / merged-lgr-1 - LGR Proposed for Ethiopic Script|Proposed LGR for Lao|Tha...

📁 Import 📄 New ⚙️ Tools ▾ [Switch mode](#) ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 🏷️ Label forms ★ About

Code points [References](#) [Meta data](#) [Tags](#) [Rules](#) [Embedded LGRs](#)

Show 500 entries

Search:

Code point	Character Name	Tags	Comm	Action
U+0E01 (n) 0 Variant(s)	THAI CHARACTER KO KAI	Thai,und-Thai-cons		See code point
U+0E02 (๓) 0 Variant(s)	THAI CHARACTER KHO KHAI			See code point
U+0E03 (๓) 0 Variant(s)	THAI CHARACTER KHO KHUAT			See code point
U+0E04 (๓) 0 Variant(s)	THAI CHARACTER KHO KHWAI			See code point
U+0E05 (๓) 0 Variant(s)	THAI CHARACTER KHO KHON			See code point
U+0E06 (๓) 0 Variant(s)	THAI CHARACTER KHO RAKHANG	Thai,und-Thai-cons		See code point
U+0E07 (๓) 0 Variant(s)	THAI CHARACTER NGO NGU	Thai,und-Thai-cons		See code point
U+0E08 (๓) 0 Variant(s)	THAI CHARACTER CHO CHAN	Thai,und-Thai-cons		See code point
U+0E09 (๓) 0 Variant(s)	THAI CHARACTER CHO CHING	Thai,und-Thai-cons		See code point
U+0E0A (๓) 0 Variant(s)	THAI CHARACTER CHO CHANG	Thai,und-Thai-cons		See code point
U+0E0B (๓) 0 Variant(s)	THAI CHARACTER SO SO	Thai,und-Thai-cons		See code point

View XML
Download
HTML Output

Click on “Output” and select “HTML Output” from the given options to view the HTML output of the LGR

HTML Output - Metadata - 1

Merged-Lgr-1

Metadata in LGR

LGR Version	2
Date	2017-10-21
Language(s)	und-Ethi und-Laoo und-Thai
Scope(s)	domain: .
Unicode Version	6.3.0

This document is mechanically formatted from the 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 Variant Sets
- 4 Classes, Rules and Actions
 - 4.1 Character Classes
 - 4.2 Whole label evaluation and context rules
 - 4.3 Actions
- 5 Table of References

Description

HTML Output - Metadata - 2

Label Generation Rules for the Thai Script

Metadata in LGR

Overview

This file contains Label Generation Rules (LGR) for the Thai script as would be appropriate for the Root zone. For more details on this LGR see "Proposal for a Thai Script Root Zone LGR [Proposal]"

Repertoire

In addition to the 68 code points according to Section 5 "Repertoire" in [Proposal], three sequences have been defined. The sequence U+0E4D U+0E32 was defined to replace the disallowed U+0E33 (THAI CHARACTER SARA AM) and to facilitate implementation of WLE rule **follows-consonant-tone** as a context rule. The other two sequences were defined to restrict U+0E45 (THAI CHARACTER LAKKHANGYAO) from appearing in any context other than these sequences. Accordingly, while U+0E45 is not listed by itself it brings the total of distinct code points to 69.

Variants

According to Section 6 "Variants", in "[Proposal]", this LGR defines no variants.

Character Classes

The Thai Script is an abugida in which consonant–vowel sequences are written as a unit: each unit is based on a consonant letter, and vowel, tone mark or diacritic notation are secondary. It is written with the combining marks stacked above or below the base consonant, like diacritics in European languages. However, although the concepts are quite similar, the implementations are significantly different.

There are 44 characters that are classified as consonants, code points from this subset have been given the tag "cons".

The 18 vowel symbols pronounced after a consonant are non-sequential: they can be located before (lv) , after (fv) , above (av) or below (bv) the consonant, or in a combination of these positions, code points from this subset have been given the tag "fv1", "fv2", "fv3", "av", "bv", "lv". There are three code point sequences defined that include vowels. (Code point sequences do not carry tag values; instead, for code point sequences the subset values are identified in comments).

There are 5 phonemic tones: mid, low, falling, high, and rising. These 5 tones are represented by 4 tone marks plus the absence of a mark. Code

HTML Output - Repertoire

Repertoire

Summary

Number of elements in repertoire	434
Number of ranges in repertoire	0
Number of code point sequences	4

Repertoire section
in LGR

Repertoire by Code Point

The following table lists the repertoire by code point (or code point sequence). The data in the Script and Name column are extracted from the Unicode character database. Where the comment in the original LGR is equal to the character name, it has been suppressed.

For any code point or sequence for which a variant is defined, the link to the associated variant set, or if mapped to itself, the variant type of that mapping is provided in the Variants column.

#	Code Point	Glyph	Script	Name	Tags	Required Context	Variants	Comment	References
1	U+0E01	ก	Thai	THAI CHARACTER KO KAI	Thai,und-Thai-cons				[5] , [100] , [101]
2	U+0E02	ข	Thai	THAI CHARACTER KHO KHAI	Thai,und-Thai-cons				[5] , [100] , [101]
3	U+0E03	ฃ	Thai	THAI CHARACTER KHO KHUAT	Thai,und-Thai-cons				[5] , [100] , [101]
4	U+0E04	ค	Thai	THAI CHARACTER KHO KHWAI	Thai,und-Thai-cons				[5] , [100] , [101]
				THAI					[5] , [100]

HTML Output - Variant Sets

Variant Sets

Summary

Number of variant sets	30
Largest variant set	4
Ordinary Variants by Type	blocked (98)

Variant Sets
section in LGR

The following tables list all variant sets defined in this LGR, except for singleton sets. Each table lists all variant mapping pairs of the set; one per row. Mappings are assumed to be symmetric: each row documents both forward (→) and reverse (←) mapping directions. In each table, the mappings are sorted by Source value in ascending code point order; shading is used to group mappings from the same source code point or sequence.

Where the *type* of both forward and reverse mappings are the same, a single value is given in the Type(s) column, otherwise the types for forward and reverse mappings, as well as comments and references are listed above one another.

A mapping where source and target are the same is *reflexive*. Variant sets consisting of only a single reflexive mapping are not shown as a set. Instead, the variant type of the mapping is listed in the Variants column of the Repertoire by Code Point table. Reflexive mappings that are part of a larger set are indicated with a “=”.

In any LGR with variant specifications that are well behaved, all members within each variant set are defined as variants of each other; the mappings in each set are symmetric and transitive; and all variant sets are disjoint.

Common Legend

Source: Source of the mapping pair.

Target: Destination of the mapping pair.

Glyph: The shape displayed for **source** or **target** depends on the fonts available to your browser.

→ - **forward:** Indicates that variant Type, References and Comment apply to the mapping from **source** to **target**.

← - **reverse:** Indicates that variant Type, References and Comment apply to the reverse mapping from **target** to **source**.

↔ - **both:** Indicates that variant Type, References and Comment apply to **both** forward and reverse mapping.

■ - **reflexive:** Indicates that variant Type, References and Comment are for a reflexive mapping where source **equals** target.

□ - **not in LGR:** Indicates that variant is **not** in LGR.

Type: The type of the variant mapping. There are some predefined variant types such as “allocatable” and “blocked”, while others are defined specifically for each LGR.

References: One or more reference IDs (optional). A “/” separates references for reverse / forward mappings, if different.

Comment: A descriptive comment (optional). A “/” separates comments for reverse / forward mappings, if different.

Variant Set 1 — 3 Members - 3 Mappings

#	Source	Glyph	Target	Glyph		Type(s)	References	Comment
1	U+1200	ᵀ	U+1210	ᵀ	↔	blocked	[4]	
2	U+1200	ᵀ	U+1280	ᵀ	↔	blocked	[4]	
3	U+1210	ᵀ	U+1280	ᵀ	↔	blocked	[4]	

HTML Output - Classes

Classes, Rules and Actions

Character Classes

Character classes
in LGR

The following table lists all top-level classes with their definition and the regular expression defining their members.

Name	Definition	Count	Members	References	Comment
und-Lao-Cf	Tag= und-Lao-Cf	14	{U+0E81 U+0E87 U+0E8A U+0E8D U+0E94 U+0E97 U+0E99 U+0E9A U+0E9F U+0EA1 U+0EA3 U+0EA5 U+0EA7 U+0EAA}		
und-Lao-consonant	Tag= und-Lao-consonant	27	{U+0E81 U+0E82 U+0E84 U+0E87 U+0E88 U+0E8A U+0E8D U+0E94 U+0E95 U+0E96 U+0E97 U+0E99 U+0E9A U+0E9B U+0E9C ...}		
und-Lao-semi-consonant	Tag= und-Lao-semi-consonant	1	{U+0EBC}		
und-Lao-tone-mark	Tag= und-Lao-tone-mark	4	{U+0EC8 U+0EC9 U+0ECA U+0ECB}		
und-Lao-vowel-above	Tag= und-Lao-vowel-above	7	{U+0EB1 U+0EB4 U+0EB5 U+0EB6 U+0EB7 U+0EBB U+0ECD}		
und-Lao-vowel-below	Tag= und-Lao-vowel-below	2	{U+0EB8 U+0EB9}		
und-Thai-above-vowel	Tag= und-Thai-av	5	{U+0E31 U+0E34 U+0E35 U+0E36 U+0E37}		Any above vowel
und-Thai-below-vowel	Tag= und-Thai-bv	2	{U+0E38 U+0E39}		Any below vowel
und-Thai-c-av-bv	([: und-Thai-consonant :]v[: und-Thai-above-vowel :]v[: und-Thai-below-vowel :])	51	{U+0E01 U+0E02 U+0E03 U+0E04 U+0E05 U+0E06 U+0E07 U+0E08 U+0E09 U+0E0A U+0E0B U+0E0C U+0E0D U+0E0E U+0E0F ...}		Any consonant, vowel-above or vowel-below
und-Thai-consonant	Tag= und-Thai-cons	44	{U+0E01 U+0E02 U+0E03 U+0E04 U+0E05 U+0E06 U+0E07 U+0E08 U+0E09 U+0E0A U+0E0B U+0E0C U+0E0D U+0E0E U+0E0F ...}		Any Consonant
und-Thai-ct	([: und-Thai-consonant :]v[: und-Thai-tone :])	48	{U+0E01 U+0E02 U+0E03 U+0E04 U+0E05 U+0E06 U+0E07 U+0E08 U+0E09 U+0E0A U+0E0B U+0E0C U+0E0D U+0E0E U+0E0F ...}		Any consonant or tone
und-Thai-ctaa	([: und-Thai-consonant :]v[: und-Thai-tone :]v[: und-Thai-sara-aa :])	49	{U+0E01 U+0E02 U+0E03 U+0E04 U+0E05 U+0E06 U+0E07 U+0E08 U+0E09 U+0E0A U+0E0B U+0E0C U+0E0D U+0E0E U+0E0F ...}		Any consonant, tone or sara-aa
und-Thai-sara-aa	Tag= und-Thai-sara-aa	1	{U+0E32}		SARA AA
und-Thai-tone	Tag= und-Thai-tone	4	{U+0E48 U+0E49 U+0E4A U+0E4B}		Any tone mark

HTML Output - Rules

Whole Label evaluation and context rules in LGR

Whole label evaluation and context rules

The following table lists all the top-level, or named rules defined in the LGR and indicates whether they are used as trigger in an action or as context (when or not-when) for a code point. (Any use of context rules for variants is not indicated).

Name	Regular Expression	Used as Trigger	Used as Context	Anchor	References	Comment
Common-leading-combining-mark	(start) ([:class property:gc=Mn:] [:class property:gc=Mc:])	True	False	False		None
und-Laoo-follows-consonant	([:und-Laoo-consonant:])← ⚓	False	True	True		WLE Rule No. 1; semi-consonant must follow a consonant
und-Laoo-precedes-consonant	⚓ →[:und-Laoo-consonant:]	False	True	True		WLE Rule No. 2; vowel-before precedes a main consonant cluster
und-Laoo-follows-main-consonant	(([:und-Laoo-consonant:] [:und-Laoo-semi-consonant:]))← ⚓	False	True	True		WLE Rule No. 3; vowel-above, and vowel-below follow a main consonant C
und-Laoo-follows-C-tonemark-vabove	(([:und-Laoo-consonant:] [:und-Laoo-semi-consonant:] [:und-Laoo-tone-mark:] [:und-Laoo-vowel-above:]))← ⚓	False	True	True		WLE Rule No. 4; vowel-after follows a main consonant, tone-mark or vowel-above
und-Laoo-consonant-cluster	([:und-Laoo-consonant:]){1,2} ([:und-Laoo-semi-consonant:]){0,1}	False	False	False		Defining consonant cluster for Rule No. 5
und-Laoo-follows-vbefore-consonant-cluster	(U+0EC0[:und-Laoo-consonant-cluster:])← ⚓	False	True	True		WLE Rule No. 5; The sequence (0EB2 0EB0) follows a vowel before, and a consonant cluster
und-Laoo-follows-C-vabove-vbelow	(([:und-Laoo-consonant:] [:und-Laoo-semi-consonant:] [:und-Laoo-vowel-above:] [:und-Laoo-vowel-below:]))← ⚓	False	True	True		WLE Rule No. 6; A tone-mark follows a main consonant, vowel-above or vowel-below
und-Laoo-follows-Cf	([:und-Laoo-Cf:])← ⚓	False	True	True		WLE Rule No. 7; The sign 0ECC can only occur after final consonants
und-Laoo-repetition-mark-limit	⚓ →((U+0EC6){0,2}(end))	False	True	True		WLE Rule No. 8; The sign 0EC6 can only occur 0 to 3 times at the end of the label
und-Thai-precedes-consonant	⚓ →[:und-Thai-consonant:]	False	True	True		WLE 7.2: check if current cp is preceding a consonant
und-Thai-follows-consonant	([:und-Thai-consonant:])← ⚓	False	True	True		WLE 7.3: check if current cp is following a consonant

HTML Output - Actions

Actions in LGR

Actions

The following table lists the actions that are used to assign dispositions to labels and variant labels, based on the specified conditions. The order of actions defines their precedence: the first action triggered by a label is the one defining its disposition.

#	Condition	Rule / Variant Set	Disposition	References	Comment
1	if label match	Common-leading-combining-mark	→ invalid		
2	if at least one variant is in	{out-of-repertoire-var}	→ invalid		any variant label with a code point out of repertoire is invalid
3	if label match	Common-leading-combining-mark	→ invalid		labels must not commence with a combining mark ◌
4	if at least one variant is in	{out-of-repertoire-var}	→ invalid		any variant label with a code point out of repertoire is invalid ◌
5	if label match	Common-leading-combining-mark	→ invalid		
6	if at least one variant is in	{out-of-repertoire-var}	→ invalid		any variant label with a code point out of repertoire is invalid

Legend

{...} - variant type set: In the "Rule/Variant Set" column the notation {...} means a set of variant types.

Table of References

- [0] The Unicode Standard 1.1, The Unicode Consortium, Mountain View, CA. 1993
- [1] The Online Encyclopedia of Writing Systems & Languages, <http://www.omniglot.com/writing/amharic.htm>, <http://www.omniglot.com/writing/argobba.htm>, <http://www.omniglot.com/writing/awngi.htm>, <http://www.omniglot.com/writing/harari.htm>, <http://www.omniglot.com/writing/xamtanga.htm>, <http://www.omniglot.com/writing/silte.htm>, <http://www.omniglot.com/writing/tigre.htm>, <http://www.omniglot.com/writing/tigrinya.htm>
Ethiopic Script Versions for the Eight Languages
- [2] Corpus Analysis performed by crawling 598 html and 40 PDF files with Tigrigna Contents published online, August 2016
Cited as Auxiliary Evidence for Tigrigna Code Points
- [3] Corpus Analysis performed by crawling 14,850 html Amharic Contents of size 1.8 GB published online, August 2016
Cited as Auxiliary Evidence for Amharic Code Points
- [4] Daniel Yacob " Application of the Double Metaphone Algorithm to Amharic Orthography", International Conference of Ethiopian Studies XV, <https://pdfs.semanticscholar.org/2f71/033d74d2f17a9502867e4a43dc4374500726.pdf>
Cited for Amharic-Driven Variants in Ethiopic Script
- [5] The Unicode Standard 1.1
- [100] Thai Industrial Standard (TIS) 1566-2541(1988) (<http://www.ratchakitcha.soc.go.th/DATA/PDF/2542/E/088/9.PDF>)
- [101] Computers and the Thai Language (http://lexicon.net.au/TH/KM_HL_5001/file_HL_5001/Paper/Inter%20Journal/km_52085.pdf)

HTML Output - Table of References

Table of References
in LGR

Table of References

- [0] The Unicode Standard 1.1, The Unicode Consortium, Mountain View, CA. 1993
- [1] The Online Encyclopedia of Writing Systems & Languages, <http://www.omniglot.com/writing/amharic.htm>, <http://www.omniglot.com/writing/argobba.htm>, <http://www.omniglot.com/writing/awngi.htm>, <http://www.omniglot.com/writing/harari.htm>, <http://www.omniglot.com/writing/xamtanga.htm>, <http://www.omniglot.com/writing/silte.htm>, <http://www.omniglot.com/writing/tigre.htm>, <http://www.omniglot.com/writing/tigrinya.htm>
Ethiopic Script Versions for the Eight Languages
- [2] Corpus Analysis performed by crawling 598 html and 40 PDF files with Tigrigna Contents published online, August 2016
Cited as Auxiliary Evidence for Tigrigna Code Points
- [3] Corpus Analysis performed by crawling 14,850 html Amharic Contents of size 1.8 GB published online, August 2016
Cited as Auxiliary Evidence for Amharic Code Points
- [4] Daniel Yacob " Application of the Double Metaphone Algorithm to Amharic Orthography", International Conference of Ethiopian Studies XV, <https://pdfs.semanticscholar.org/2f71/033d74d2f17a9502867e4a43dc4374500726.pdf>
Cited for Amharic-Driven Variants in Ethiopic Script
- [5] The Unicode Standard 1.1
- [100] Thai Industrial Standard (TIS) 1566-2541(1988) (<http://www.ratchakitcha.soc.go.th/DATA/PDF/2542/E/088/9.PDF>)
- [101] Computers and the Thai Language (http://lexitron.nectec.or.th/KM_HL5001/file_HL5001/Paper/Inter%20Journal/krrn_52085.pdf)
- [201] Lao grammar book published by the Ministry of Education in 1967, see Appendix B, Figure 1
- [202] Lao grammar book published by the Ministry of Education in 1967, see Appendix B, Figure 2
- [203] Lao grammar book published by the Ministry of Education in 1967, see Appendix B, Figure 3

Validate a Label

Validate a Label

The screenshot shows the LGR Tool interface for 'proposal-lao-lgr-31jan17-en - Proposed LGR for Lao'. The top navigation bar includes buttons for 'Import', 'New', 'Tools', 'Switch mode', 'Validate label', 'Validate LGR', 'Output', 'Label forms', and 'About'. The 'Validate label' button is highlighted with a checkmark. Below the navigation bar, there are tabs for 'Code', 'References', 'Meta data', 'Tags', and 'Rules'. A table of Lao letters is displayed, with columns for 'Code', 'Name', 'Tags', and 'Action'. Three callout boxes provide instructions: 1. Click on 'Import' to load single or multiple existing LGR files. 2. Always view 'Validate LGR' output as the tool checks the loaded LGR(s) during this process. 3. For validating a label, click on 'Validate label' button.

1. Click on "Import" to load single or multiple existing LGR files. See "Import" for details

2. Always view "Validate LGR" output as the tool checks the loaded LGR(s) during this process

3. For validating a label, click on "Validate label" button

Code	Name	Tags	Action
<input type="checkbox"/>	U+0E87 (๗) 0 Variant(s)		See code point
<input type="checkbox"/>	U+0E88 (๘) 0 Variant(s)		See code point
<input type="checkbox"/>	U+0E8A (๘) 0 Variant(s)	Cf,consonant	See code point
<input type="checkbox"/>	U+0E8D (๘) 0 Variant(s)	Cf,consonant	See code point
<input type="checkbox"/>	U+0E94 (๗) 0 Variant(s)	Cf,consonant	See code point
<input type="checkbox"/>	U+0E95 (๗) 0 Variant(s)	consonant	See code point

Validate a Label with a Single LGR

1. Enter the label to be validated

2. Click on the "Validate" button

3. Check to include mix script variants

Label: بكلك

Maximum length: 63 code points

Include mixed-script variants

Validate

Code points

→ Select code p

<input type="checkbox"/>	Code
<input type="checkbox"/>	U+0021 0 Variant
<input type="checkbox"/>	U+0030 2 Variant
<input type="checkbox"/>	U+0031 2 Variant
<input type="checkbox"/>	U+0032 2 Variant
<input type="checkbox"/>	U+0033 2 Variant

Action

See code point

See code point

See code point

See code point

See code point

Validate a Label with a Single LGR

Result given by “Validate label” shows whether the label is valid or invalid and also lists its variants, their dispositions and the rules/actions against which the label or any of its variants is valid or invalid

Validate label

Label: (Maximum length: 63 code points)

✓ VALID

U-label	Disposition	Code point sequence	
ເທດທາງ xn-- p6c1bq3etb6e	valid	U+0EC0 (໒) U+0EAB (໓) U+0E94 (໔) U+0E81 (໕) U+0EB2 (໖) U+0E99 (໗)	Show / hide log

[Action index: 4] <action disp="valid" comment="catch all e"/>

Variant labels (including original as last)

1 variant label(s) generated.
By disposition: Counter({'valid': 1})

ເທດທາງ xn-- p6c1bq3etb6e	valid	U+0EC0 (໒) U+0EAB (໓) U+0E94 (໔) U+0E81 (໕) U+0EB2 (໖) U+0E99 (໗)	Show / hide log
--------------------------------	-------	---	-----------------

[Action index: 4] <action disp="valid" comment="catch all e"/>

Validate a Label with a Merged LGR

Validate label

Script: und-Laoo
The script used to validate the label

✓ VALID

U-label	Disposition	Code point sequence	
ဗေတကူ xn-- p6c1bq3etb6e	valid	U+0EC0 (၄) U+0EAB (၈) U+0E94 (၁) U+0E81 (၈) U+0EB2 (၇) U+0E99 (၁)	Show / hide log

[Action index: 4] <action disp="valid" comment="catch all *"/>

✓ No collision

Collision

No collision.

Variant labels (including original as last)

1 variant label(s) generated.
By disposition: Counter({'valid': 1})

ဗေတကူ xn-- p6c1bq3etb6e	valid	U+0EC0 (၄) U+0EAB (၈) U+0E94 (၁) U+0E81 (၈) U+0EB2 (၇) U+0E99 (၁)	Show / hide log
-------------------------------	-------	---	-----------------

Result given by "Validate label" shows whether the label is valid or invalid and also lists its variants, their dispositions and the rules/actions of which the label or any of its variants is valid or invalid

Review Impact on Existing Labels by Revising an Existing LGR

Review Impacts on Existing Labels Caused by Reviewed LGR

LGR Tool / proposed-arabic-lgr-18092016-en - Proposed LGR for Arabic Script

Import New Tools Switch mode Validate label Validate LGR Output Label forms About

Code points Refer

Select code point(s) to

- Compare two LGRs
- Diff labels of two LGRs**
- Get collisions in a list of labels
- Cross-script variants
- Generate disp. annotations
- Harmonization

entries

code point(s)

<input type="checkbox"/>	Code point				
<input type="checkbox"/>	U+0620 () 0 Variant(s)		sc:Arab		See code point
<input type="checkbox"/>	U+0621 (*) 0 Variant(s)	ARABIC LETTER HAMZA	sc:Arab		See code point
<input type="checkbox"/>	U+0622 (!) 4 Variant(s)	ARABIC LETTER ALEF WITH MADDA ABOVE	sc:Arab		See code point
<input type="checkbox"/>	U+0623 (!) 4 Variant(s)	ARABIC LETTER ALEF WITH HAMZA ABOVE	sc:Arab		See code point
<input type="checkbox"/>	U+0624 (z) 1 Variant(s)	ARABIC LETTER WAW WITH HAMZA ABOVE	sc:Arab		See code point
<input type="checkbox"/>	U+0625 (!) 4 Variant(s)	ARABIC LETTER ALEF WITH HAMZA BELOW	sc:Arab		See code point
<input type="checkbox"/>	U+0626 (sz) 7 Variant(s)	ARABIC LETTER YEH WITH HAMZA ABOVE	sc:Arab		See code point
<input type="checkbox"/>	U+0627 (!) 4 Variant(s)	ARABIC LETTER ALEF	sc:Arab		See code point

Click on "Diff labels of two LGRs" button to determine differences caused by modifying an LGR

Enter Details

🏠 LGR Tool / proposed-arabic-lgr-18092016-en - Proposed LGR for Arabic Script

📁 Import 📄 New ⚙️ Tools ▾

Switch mode ✓ Validate label 📄 Validate LGR ↻ Output ▾ 📄 Label forms ★ About

First LGR **1. Select first LGR**
First LGR to use in diff

Second LGR **2. Select second LGR**
Second LGR to use in diff

Labels test-labels.txt **3. Select file containing labels**
List of labels to use in diff. File must be encoded in UTF-8 and using UNIX line ending.

Check collisions
Also check for collision of labels in both LGR

Output rules
Show rules in output (this can be very verbose)

5. Check collisions if you want to check label collisions as well







6. Check "Output rules" if you want to check output rules for each label

7. Click on "Get diff" button

Download Results from Task Status Page

Tasks

Tasks are ordered from newest to latest

Annotate labels on LGR 5	July 25, 2023, 5:05 p.m.	 Download report	Success		The report expires in 15 days
Review of 1 IDN table	July 21, 2023, 6:57	 Download report	Success		The report expires in 11 days
Review of 1 IDN table	July 1	 Download report	Success		The report expires in 3 days
Review of 1 IDN table	July 1	 Download report	Success		The report expires in 3 days

1. Click on the download link on the homepage to get the “Diff labels of two LGRs” results

When you delete a task, the corresponding report is not deleted

 Delete completed tasks

Download Results

Advanced LGR (Label Generation)

1. Click on Home link for LGR Tool to get to this page

This application provides a convenient interface for browsing and editing LGRs conforming to the [Representing Label Generation Rulesets using XML](#) specification.

Previously loaded LGR file(s)

Previously, you edited the following LGR file(s). Click on its title to resume your editing session.

LGRs

- [View Demo-1](#) 🗑️
- [View lgr-second-level-arabic-language-31may22-en](#) 🗑️

Create a new LGR file or import an existing one

📁 Import an existing XML file

⚠️ **Note that importing large LGR files may take significant time to load on your browser.**

📄 Start with a New blank XML file

Start from a built-in LGR

The following LGRs are pre-installed in the system. You may use them as a starting point for your own LGR. To do this, click on its title to make a copy that you can then edit.

Your saved results

The following files contains your tools computation results.

⚠️ **Note that these files could be cleaned up regularly.**

- [Download 20221022_171249_annotation_RZ-LGR_5.txt.gz](#) 🗑️
- [Download 20221022_174606_diff_Demo-1_lgr-second-level-arabic-language-31may22-en.txt.gz](#) 🗑️

2. Click on the download link on the homepage to get the “Diff labels of two LGRs” results

Results

```
# Labels not in LGR 1 #
```

```
# Labels not in LGR 2 #
```

```
Label عملية
```

```
Label نادي
```

```
# LGR comparison #
```

```
Label عملية not in LGR proposed-arabic-lgr-18092017-en
```

```
Label نادي not in LGR proposed-arabic-lgr-18092017-en
```

```
## Comparison on label 'كلكتيم' [U+06A9 U+0644 U+0643 U+062A U+06C1]
```

```
### Test dispositions: ###
```

```
^^^
```

```
No changes in disposition.
```

```
^^^
```

```
### Test number of variants: ###
```

```
^^^
```

```
No changes in number of variants.
```

```
^^^
```

```
## Comparison on label 'لقاء' [U+0644 U+0642 U+0627 U+0621]
```

```
### Test dispositions: ###
```

```
^^^
```

```
No changes in disposition.
```

```
^^^
```

```
### Test number of variants: ###
```

```
^^^
```

```
No changes in number of variants.
```

```
^^^
```

Specifies changes in disposition of labels by the revised LGR

Specifies new variants of labels formed by the revised LGR

Review Impact on Existing Labels by Introducing a New LGR

Select Get Collisions in a List of Labels

LGR Tool / proposed-arabic-lgr-18092016-en - Proposed LGR for Arabic Script

Import New Tools

Switch mode Validate label Validate LGR Output Label forms About

Code points Referer

Expand range(s) Add code point(s)

Select code point(s) to

Search:

Get collisions in a list of labels

Compare two LGRs

Diff labels of two LGRs

Cross-script variants

Generate disp. annotations

Harmonization

<input type="checkbox"/>	Code point		Tags	Comments	Action
<input type="checkbox"/>	U+0620 () 0 Variant(s)				See code point
<input type="checkbox"/>	U+0621 (◌) 0 Variant(s)	ARABIC LETTER HAMZA			See code point
<input type="checkbox"/>	U+0622 (ﻯ) 4 Variant(s)	ARABIC LETTER ALEF WITH MADDA ABOVE			See code point
<input type="checkbox"/>	U+0623 (ﻰ) 4 Variant(s)	ARABIC LETTER ALEF WITH HAMZA ABOVE			See code point
<input type="checkbox"/>	U+0624 (ﻲ) 1 Variant(s)	ARABIC LETTER WAW WITH HAMZA ABOVE			See code point
<input type="checkbox"/>	U+0625 (ﻲ) 4 Variant(s)	ARABIC LETTER ALEF WITH HAMZA BELOW	sc:Arab		See code point
<input type="checkbox"/>	U+0626 (ﻲ) 7 Variant(s)	ARABIC LETTER YEH WITH HAMZA ABOVE	sc:Arab		See code point
<input type="checkbox"/>	U+0627 (ﺀ) 4 Variant(s)	ARABIC LETTER ALEF	sc:Arab		See code point

Click on “Get collisions in a list of labels” button to determine label collisions from an existing file if a new LGR is introduced – for example, two unique labels become variants of each other

Enter Details

LGR

lgr-second-level-arabic-language-31may22-en

LGR to use in tool

1. Select LGR

Also check for collision with existing TLDs

2. Select Labels file

Choose File No file chosen

3. Check with existing TLDs

Labels must be encoded in UTF-8 and using 0x0A line ending.

See the [FAQ](#) for more information on the [task status page](#).

Full dump

Print a full dump

4. Check "Full Dump" to get summary of each operation done on the labels

⚠ Note that collision


Get collisions

6. Click on "Get collisions"

Download Results from Task Status Page

Tasks


Review of 1 IDN table - Oct. 20, 2022, 10:35 a.m.

 Download report

Success




Annotate labels on LGR 5 - Oct. 22, 2022, 5:12 p.m.

 Download report

Success




Diff with LGR Demo-1 - Oct. 22, 2022, 5:46 p.m.

 Download report

Success




Collision with LGR lgr-second-level-arabic-language-31may22-en - Oct. 22, 2022, 5:52 p.m.

 Download report

Success



 Delete completed tasks

Click on the download link to
get the “Collisions” results

Download Results

🏠 LGR Tools

📁 Import

📄 New

⚙️ Tools ▾

Click on Home link for LGR Tool to get to this page

Switch mode

📄 Label forms

★ About

Advanced LGR (Label Generation Ruleset) Tools

This application provides a convenient interface for browsing and editing [LGRs](#) conforming to the [Representing Label Generation Rulesets using XML](#) specification.

Previously loaded LGR file(s)

Previously, you edited the following LGR file(s). Click on its title to resume your editing session.

LGRs

- [View lgr-new](#) 🗑️
- [View sample-french](#) 🗑️
- [View union-of-sample-french-and-lgr-4-arabic-script-29jun20-en](#) 🗑️

Your saved results

The following files contains your tools computation results.

⚠️ Note that these files could be cleaned up regularly.

- [Download 20210528_102129_collisions_lgr-4-arabic-script-29jun20-en.txt.gz](#) 🗑️

Create a new LGR file or import an existing one

📁 Import an existing XML file

⚠️ Note that importing large LGR files may take significant time to load on your browser.

📄 Start with a New blank XML file

Start from a built-in LGR

The following LGRs are pre-installed in the system. You may use them as a starting point for your own LGR. To do so,

Click on the download link to get the “Collisions” results

Results

Labels not in LGR

Label مجموعة
Label بهجده ب
Label مرفق
Label مفعول
Label معضلة
Label لجنة
Label محمد
Label موعده
|

Collisions

Collision

```
***  
Label:      'علكتكم' | 'علكتكم'  
Code points: [U+06AA U+0644 U+0643 U+062A U+06C1] | [U+06A9 U+0644 U+0643 U+062A U+06C1]  
Category:   Primary | Primary  
***
```

Collision

```
***  
Label:      'علكتة' | 'علكتكم'  
Code points: [U+06AA U+0644 U+0643 U+062A U+06C1] | [U+06AA U+0644 U+0643 U+062A U+0629]  
Category:   Primary | Variant  
***
```

Details for label 'علكتكم' [U+06AA U+0644 U+0643 U+062A U+06C1]

```
***  
Variant 'علكتة' [U+06AA U+0644 U+0643 U+062A U+0629]:  
Disposition: invalid  
Rules:  
***
```

Details for label 'علكتكم' [U+06A9 U+0644 U+0643 U+062A U+06C1]

```
***  
Variant 'علكتة' [U+06AA U+0644 U+0643 U+062A U+0629]:  
Disposition: invalid
```

List of labels which have become invalid by introducing a new LGR

Details of a label and its corresponding variant(s)

Validate Multiple Labels Using a Single or Merged LGR(s)

Select Generate Disposition Annotations

🏠 LGR Tools /sample-french - Sample LGR for French

📁 Import 📄 New ⚙️ Tools ▾ Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 📄 Label forms ★ About

- 🔄 Diff labels of two LGRs
- 🔍 Get collisions in a list of labels
- 🔄 Cross-script variants
- 📄 **Generate disp. annotations**
- ✍️ Compute variants
- 📊 Harmonization

1. Click on "Import" to load existing LGR file(s). See "Import" for details

2. Click on "Generate disp. annotations" button to validate labels given in a text file

Expand range(s) Add code point(s)

Search:

<input type="checkbox"/>	Code point	Label Name	Tags	Comments	Action
<input type="checkbox"/>	U+0061 (a) 0 Variant(s)	ALL LETTERS			See code point
<input type="checkbox"/>	U+0062 (b) 0 Variant(s)	LATIN SMALL LETTER B			See code point
<input type="checkbox"/>	U+0063 (c) 0 Variant(s)	LATIN SMALL LETTER C			See code point
<input type="checkbox"/>	U+0064 (d) 0 Variant(s)	LATIN SMALL LETTER D			See code point

Enter Details for Single LGR

🏠 LGR Tools

📁 Import

📄 New

⚙️ Tools ▾

[Switch mode](#)

Tasks

Help ▾

📄 Label forms

👤 lgr-user-saqib lgr-user-saqib ▾

LGR

lgr-second-level-arabic-language-31may22-en

1. Select LGR

LGR to use in tool

Labels

Choose File No file chosen

2. Select Labels file for validation. The labels file contains single label per line in UTF-8 format

List of labels to use in tool. File must be encoded in UTF-8 and using 0x0A line ending.

The computing may be very long, you can follow your task progression on the [task status page](#).

Annotate

If you have questions, please contact globalsupport@icann.org

English (en) ▾

Go

3. Click on "Annotate" button

Enter Details for Merged LGR

LGR Tools /lgr-set - Root Zone LGR for Devanagari|Root Zone LGR for Ethiopic|Roo...

Import

New

Tools

Switch mode

Validate label

Validate LGR

Output

Label forms

About

LGR

lgr-set

1. Select LGR

LGR to use in tool

Allocated Set labels

Choose File No file chosen

Optional file of existing labels to check for collisions

Optional list of labels already allocated in the LGR set, that will be used to check for collisions when... and using 0x0A line ending.

Script

und-Geor

2. Select a script from the list of the scripts of different LGRs forming the merged LGR

The script used to validate the label

Labels

Choose File malayalam-test-labels-26jun20-en.txt

3. Select Labels file for validation

List of labels to use in tool. File must be encoded in UTF-8 and using 0x0A line ending.

Provide your e-mail address






Annotate

4. Click on "Annotate" button

Download Results from Task Status Page

Tasks

Tasks are ordered from newest to latest

Annotate labels on LGR 5	July 25, 2023, 5:05 p.m.	 Download report	Success		The report expires in 15 days
Review of 1 IDN table	July 21, 2023, 6:57 a.m.				The report expires in 11 days
Review of 1 IDN table	July 13, 2023, 2:56 p.m.	 Download report	Success		The report expires in 3 days
Review of 1 IDN table	July 13, 2023, 2:55 p.m.	 Download report	Success		The report expires in 3 days

Click on the download link to get the “Generate disp. annotations” results

When you delete a task, the corresponding report is not deleted

 Delete completed tasks

Download Results

Home LGR Tools

Import New Tools

1. Click on Home link for LGR Tool to get to this page

Switch mode Label forms About

Advanced LGR (Label Generation Ruleset) Tools

This application provides a convenient interface for browsing and editing [LGRs](#) conforming to the [Representing Label Generation Rulesets using XML](#) specification.

Previously loaded LGR file(s)

Previously, you edited the following LGR file(s). Click on its title to resume your editing session.

LGRs

- [View lgr-new](#)
- [View sample-french](#)
- [View union-of-sample-french-and-lgr-4-arabic-script-29jun20-en](#)

Your saved results

The following files contains your tools computation results.

⚠ Note that these files could be cleaned up regularly.

- [Download 20210528_102129_collisions_lgr-4-arabic-script-29jun20-en.txt.gz](#)

Create a new LGR file or import an existing one

Import an existing XML file

⚠ Note that importing large LGR files may take significant time to load on your browser.

Start with a New blank XML file

Start from a built-in LGR

The following LGRs are pre-installed in the system. You may use them as a starting point for your own LGR. To do so,

Click on the download link to get the “Generate disp. annotations” results

Results

```
ທັງໃຕ: valid
ໄປນອກ: valid
ເຂືອຈັກ: valid
ຕີນສີແດງ: valid
ຈໍາເຈ: invalid
  Code point U+0EB3 not in repertoire
ຫຼຸມແພງ: valid
ອະນຸດຕະສະ: valid
ສາຍເລືອດແດງ: valid
ສິນເຫດ: valid
ລູກແຕ່ເຊົາ: valid
ສຸສະ: valid
ສະຈິດ: valid
ເຂັ້ວບໍ່ໄຫວ: valid
ນັກພິສິກ: valid
ດັງຄາດ: valid
ລະເຫລີງ: valid
ເປັນຕາຂັກ: valid
ໄໝ້ງ: invalid
  Code point U+0EC2 does not comply with rules 'precedes-consonant'
  Code point U+0EDD not in repertoire
  Code point U+0EC9 does not comply with rules 'follows-C-vabove-vbelow'
ເຫາະ: valid
ພະສູ່ເປັນເຈົ້າ: valid
ຫນຶ່ງ: valid
ທຸກຫົນທຸກແຫ່ງ: valid
ແມ່ສີ: valid
ຮົວມ້ອຍ: valid
ຫລອກລໍ່: valid
ຕາມຄວນ: valid
ສຸພາບອ່ອນຫວານ: valid
ບໍ່ຫົນ: valid
ຫມີ: valid
ແຮງຄົນ: valid
.. .. .
```

Validation result of each label is written next to it. For invalid labels, the tool also specifies the reason/rule due to which the label is invalid

View Cross-Script Variants of Labels

Select Cross-script Variants

🏠 LGR Tools /sample-french - Sample LGR for French

📁 Import 📄 New ⚙️ Tools ▾ Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 📄 Label forms ★ About

- Compare two LGRs
- 🔄 Diff labels of two LGRs
- ✂️ Get collisions in a list of labels
- 🔄 Cross-script variants
- 📄 Generate disp. annotations
- ✍️ Compute variants
- 📊 Harmonization

1. Click on “Import” to load single or multiple LGR file(s). See “Import” for details

2. Click on “Cross-script variants” button to view cross-script variants of labels given in a text file

Code points Referenc...

Expand range(s) Add code point(s)

Search:

<input type="checkbox"/>	Code point	Tags	Comments	Action
<input type="checkbox"/>	U+0061 (a) 0 Variant(s)			See code point
<input type="checkbox"/>	U+0062 (b) 0 Variant(s)	LATIN		See code point
<input type="checkbox"/>	U+0063 (c) 0 Variant(s)	LATIN		See code point
<input type="checkbox"/>	U+0064 (d) 0 Variant(s)	LATIN SMALL LETTER D		See code point

Enter Details

🏠 LGR Tool

📁 Import 📄 New ⚙️ Tools ▾

Switch mode 📄 Label forms ★ About

LGR merged-lgr ▾
LGR to use for tool

Script und-Armn ▾
The script used to validate the label

Labels Choose File test-labels.txt
List of labels to use in diff. File must be encoded in UTF-8 and using UNIX line ending.

English (en) ▾ Go

©


Launch

1. Select single LGR or a merged LGR
2. Select a script from the list of the scripts of different LGRs forming the merged LGR
3. Select Labels file for validation
4. Click on "Launch" button

Download Results from Task Status Page

Tasks


Review of 1 IDN table - Oct. 20, 2022, 10:35 a.m.

 Download report

Success




Annotate labels on LGR 5 - Oct. 22, 2022, 5:12 p.m.

 Download report

Success




Diff with LGR Demo-1 - Oct. 22, 2022, 5:46 p.m.

 Download report

Success




Collision with LGR lgr-second-level-arabic-language-31may22-en - Oct. 22, 2022, 5:52 p.m.

 Download report

Success



 Delete completed tasks

Click on the download link to get the “Cross-script variants” results

Download Results

Advanced LGR (Label Generation Ruleset) Tools

This application provides a convenient interface for browsing and editing [LGRs](#) conforming to the [Representing Label Generation Rulesets using XML](#) specification.

Previously loaded LGR file(s)

Previously, you edited the following LGR file(s). Click on its title to resume your editing session.

LGRs

- [View lgr-new](#) 🗑️
- [View sample-french](#) 🗑️
- [View union-of-sample-french-and-lgr-4-arabic-script-29jun20-en](#) 🗑️

Your saved results

The following files contains your tools computation results.

⚠️ Note that these files could be cleaned up regularly.

- [Download 20210528_102129_collisions_lgr-4-arabic-script-29jun20-en.txt.gz](#) 🗑️

Create a new LGR file or import an existing one

[Import an existing XML file](#)

⚠️ Note that importing large LGR files may take significant time to load on your browser.

[Start with a New blank XML file](#)

Start from a built-in LGR

The following LGRs are pre-installed in the system. You may use them as a starting point for your own LGR. To do so

Click on the download link to get the “Cross-script variants” results

Results

```
Input label U+0561 U+0575 U+0580 (wɯɹ̥) has cross-script variants:
- Cross-variant U+0448 U+0575 U+0580 (wɯɹ̥), disposition invalid:
  + U+0448 (w): Cyrl
Input label U+0581 U+0561 U+0579 U+0586 (gɯɯɹ̥) has cross-script variants:
- Cross-variant U+0581 U+0448 U+0579 U+0586 (gɯɯɹ̥), disposition invalid:
  + U+0448 (w): Cyrl
- Cross-variant U+0067 U+0561 U+0579 U+0586 (gɯɯɹ̥), disposition invalid:
  + U+0067 (g): Latn
- Cross-variant U+0067 U+0448 U+0579 U+0586 (gɯɯɹ̥), disposition invalid:
  + U+0448 (w): Cyrl
  + U+0067 (g): Latn
Input label U+0566 U+0578 U+057D (qnu) has cross-script variants:
- Cross-variant U+0566 U+0578 U+0075 (qnu), disposition invalid:
  + U+0075 (u): Latn
- Cross-variant U+0566 U+006E U+057D (qnu), disposition invalid:
  + U+006E (n): Latn
- Cross-variant U+0566 U+006E U+0075 (qnu), disposition invalid:
  + U+0075 (u): Latn
  + U+006E (n): Latn
- Cross-variant U+0071 U+0578 U+057D (qnu), disposition invalid:
  + U+0071 (q): Latn
- Cross-variant U+0071 U+0578 U+0075 (qnu), disposition invalid:
  + U+0071 (q): Latn
  + U+0075 (u): Latn
- Cross-variant U+0071 U+006E U+057D (qnu), disposition invalid:
  + U+0071 (q): Latn
  + U+006E (n): Latn
- Cross-variant U+0071 U+006E U+0075 (qnu), disposition invalid:
  + U+0071 (q): Latn
  + U+0075 (u): Latn
  + U+006E (n): Latn
```

Specifies the LGR (from a set of multiple LGRs) that is used to generate these cross-script variants

Cross-script variants of the label

Disposition of cross-script variants

Compare LGRs

Compare LGRs

🏠 LGR Tools /sample-french - Sample LGR for French

📁 Import 📄 New ⚙️ Tools ▾ Switch mode ✓ Validate label 📄 Validate LGR 🔄 Output ▾ 📄 Label forms ★ About

- ↔️ Compare two LGRs
- 🔍 Diff labels of two LGRs
- 🔍 Get collisions in a list of labels
- ↔️ Cross-script variants
- 📄 Generate disp. annotations
- ✍️ Compute variants
- 📊 Harmonization

Click on “Compare two LGRs” under the “Tools” tab for comparing LGRs

Code points Referer

Expand range(s) Add code point(s)

Select code point(s)

Search:

<input type="checkbox"/>	Code point	er Name	gs	Comments	Action
<input type="checkbox"/>	U+0061 (a) 0 Variant(s)	ALL LETTER A			See code point
<input type="checkbox"/>	U+0062 (b) 0 Variant(s)	LATIN SMALL LETTER B			See code point
<input type="checkbox"/>	U+0063 (c) 0 Variant(s)	LATIN SMALL LETTER C			See code point
<input type="checkbox"/>	U+0064 (d) 0 Variant(s)	LATIN SMALL LETTER D			See code point

Union, Intersection or Difference

↑ LGR Tool / proposed-arabic-lgr-18092017-en - Proposed LGR for Arabic Script

Import

New

Tools

Switch mode

Validate label

Validate LGR

Output

Label forms

About

First LGR

proposed-arabic-lgr-18092016-en

First LGR to use in comparison

1. Select first LGR

Second LGR

proposed-arabic-lgr-18092017-en

Second LGR to use in comparison

2. Select second LGR

Action to perform on LGRs

✓ Union
Intersection
Diff

Compare

⚠ Note that comparison will be performed only between two simple LGR sets or two simple LGRs, union and intersection are not available for LGR sets.

English (en) Go

3. Select "Union", "Intersection" or "Diff" to perform relevant function on the two LGRs

4. Click on the "Compare" button

With Merged LGRs, Only Difference

🏠 LGR Tool / merged-lgr-2 - Initial Release | Proposed LGR for Khmer

📁 Import

📄 New

⚙️ Tools ▾

[Switch mode](#)

✓ Validate label

📄 Validate LGR

🔄 Output ▾

🔍 Label forms

★ About

First LGR

merged-lgr-1

First LGR to use in comparison

1. Select first LGR

Second LGR

merged-lgr-2

Second LGR to use in comparison

2. Select second LGR

Action to perform on
LGRs

Diff

Choose the action to perform on selected LGRs

“Diff” selected by default

Compare

⚠️ Note that comparison will be performed only between two LGR sets or two simple LGRs, union and intersection are not available for LGR sets.

English (en) ▾

Go

3. Click on the
“Compare” button

Result of Difference function

🏠 LGR Tool / proposed-arabic-lgr-18092017-en - Proposed LGR for Arabic Script

📁 Import

📄 New

⚙️ Tools ▾

Switch mode

✓ Validate label

📄 Validate LGR

🔄 Output ▾

📄 Label forms

★ About

Result of diff of **proposed-arabic-lgr-18092016-en** with **proposed-arabic-lgr-18092017-en**

** Compare Metadata **

Compare Description:
Second LGR has no description

Languages values differ:
Values only in first LGR: ['und-Arab'].
Values only in second LGR: [].

** Compare repertoire **

Repertoire values differ:
Values only in first LGR: U+062F U+063A U+0635 U+0636 U+0637 U+0630 U+0631 U+062C U+0632 U+062D U+0638 U+0633 U+062E U+0639 U+0634.
Values only in second LGR: .

** Compare common code points in repertoire **

113 code points are identical

** Compare WLE **

Differences of two LGRs

Important Note: These operations only provide provisional results which must be manually reviewed and finalized

English (en) ▾

Go

©

Harmonize Multiple LGRs

Select Harmonization

LGR Tools / Union of Demo-1 and lgr-second-level-arabic-language-31may22-en - Second Level Reference LGR

Import New Tools

- Compare two LGRs
- Diff labels of two LGRs
- Get collisions in a list of labels
- Compute variants
- Harmonization**
- Cross-script variants
- Generate disp. annotations

Validate label Validate LGR Output Label forms lgr-user-saqib lgr-user-saqib

1. Click on "Import" to load single or multiple LGR file(s). See "Import" for details

2. Click on "Harmonization" button to harmonize two LGR files

Code points Reference

Select code point(s) to

Add code point(s)

Search:

<input type="checkbox"/>	Code point				Action
<input type="checkbox"/>	U+002D (-) 0 Variant(s)	HYPHEN-MINUS	sc		See code point
<input type="checkbox"/>	U+0030 (0) 2 Variant(s)	DIGIT ZERO	Common-digit,sc:Zyyy	DIGIT ZERO; ☺	See code point
<input type="checkbox"/>	U+0031 (1) 2 Variant(s)	DIGIT ONE	Common-digit,sc:Zyyy	DIGIT ONE; ☺	See code point
<input type="checkbox"/>	U+0032 (2) 2 Variant(s)	DIGIT TWO	Common-digit,sc:Zyyy	DIGIT TWO; ☺	See code point
<input type="checkbox"/>	U+0033 (3) 2 Variant(s)	DIGIT THREE	Common-digit,sc:Zyyy	DIGIT THREE; ☺	See code point
<input type="checkbox"/>	U+0034 (4) 2 Variant(s)	DIGIT FOUR	Common-digit,sc:Zyyy	DIGIT FOUR; ☺	See code point

Enter Details

🏠 LGR Tools

📁 Import

📄 New

⚙️ Tools ▾

Switch mode

Tasks

Help ▾

🔠 Label forms

👤 lgr-user-yinmay lgr-user-yinmay ▾

First LGR

Sample-French

First LGR to use in tool

1. Select first LGR



Second LGR

Greek Testing txt LGR

Second LGR to use in tool

2. Select second LGR



Root Zone LGR

RZ-LGR 5

The (optional) RootZone LGR to infer new variant sets from

Optional RootZone LGR
to infer new variant sets



Harmonize

3. Click on "Harmonize"
button

Results of harmonization for Sample-French and Greek Testing txt LGR

Resulting LGRs

- [View harmonized version of Sample-French_harmonized_20230808_151748](#)
- [View harmonized version of Greek Testing txt LGR_harmonized_20230808_151748](#)

Harmonized LGRs

If you have questions, please contact globalsupport@icann.org

©

The Harmonized LGRs will also be listed on the homepage

Compute Variants

Compute Variants

🏠 LGR Tools

📁 Import

📄 New

⚙️ Tools ▾

Switch mode

🔍 Label f

Advanced

This application provides a

Previously loaded

Previously, you edited the

LGRs

- [View lgr-new](#) 🗑️
- [View sample-french](#) 🗑️
- [View union-of-sample-french-and-lgr-4-arabic-script-29jun20-en](#) 🗑️

Create a new LGR file or import an existing one

📁 Import an existing XML file

↔️ Compare two LGRs

↔️ Diff labels of two LGRs

✂️ Get collisions in a list of labels

↔️ Cross-script variants

📄 Generate disp. annotations

✂️ Compute variants

📊 Harmonization

1. Click on "Import" to load single or multiple LGR file(s). See "Import" for details

2. Click on "Compute Variants" to compute variants of label(s) against imported LGR

Your saved results

The following files contains your tools results.

⚠️ Note that these files could be cleared

[Download 20210528_102129_cross-script-29jun20-en.txt.gz](#)

Enter Details

🏠 LGR Tool / lgr-4-devanagari-script-29jun20-en - Root Zone LGR for Devanagari

📁 Import

📄 New

⚙️ Tools ▾

Switch mode

✓ Validate label

📄 Validate LGR

🔄 Output ▾

🔠 Label forms

★ About

LGR

lgr-4-devanagari-script-29jun20-en

LGR to use in tool

Labels

Choose File test.txt

List of labels to use in tool. File must be encoded in UTF-8 and use UNIX line ending.

1. Imported LGR

2. File which contains labels to be used for variant computation

Compute variants

3. Press this to compute variants

⚠️ Note that variants tool is not available for LGR sets.













English (en) ▾

Go

Results

Tasks

Tasks are ordered from newest to latest

Variant computation on LGR lgr-second-level-arabic-script-31may22-en (1)	Aug. 2, 2023, 12:54 p.m.	 Download report	 Success		The report expires in 15 days
Review of 1 IDN table	July 31, 2023, 12:43 p.m.	 Download report	 Success		The report expires in 13 days
Annotate labels on LGR 5	July 25, 2023, 5:05 p.m.	 Download report	 Success		The report expires in 8 days
Review of 1 IDN table	July 21, 2023, 6:57 a.m.	 Download report	 Success		The report expires in 3 days

When you delete a task, the corresponding report is not deleted

 Delete completed tasks

Download results of
compute variants

Results

🏠 LGR Tool

Press here to go to homepage. The results will be present on the Homepage when available

📁 Import

📄 New

⚙️ Tools ▾

[Switch mode](#)

🔍 Label forms



Welcome to the LGR (Label Generation Ruleset) Tool

This application provides a convenient interface for browsing and editing LGRs conforming to the [Representing Label Generation Rulesets using XML](#) specification.

Previously loaded LGR file(s)

Previously, you edited the following LGR file(s). Click on its title to resume your editing session.

LGRs

- [View lgr-4-arabic-script-29jun20-en](#) 🗑️
- [View lgr-4-devanagari-script-29jun20-en](#) 🗑️

Create a new LGR file or import an existing one

📁 Import an existing XML file

⚠️ Note that importing large LGR files may take significant time to load on your browser.

Your saved results

The following files contains your tools computation results.

⚠️ Note that these files could be cleaned up regularly.

- [Download 20201120_092126_labels_variants_lgr-4-devanagari-script-29jun20-en.csv.gz](#) 🗑️

Download results of compute variants

Label Forms

A label, U label against IDNA2008

Label Forms

- Type in a label or upload a list of labels

Click to start using the Label forms function



↑ LGR Tools Tasks Help ▾ **Label forms** lgr-user ▾

Welcome to the LGR (Label Generation Ruleset) Tools

Select your mode:

✓ Validate label(s) against an LGR

Review IDN table(s)

Advanced LGR Tools

If you have questions, please contact globalsupport@icann.org

English (en) ▾ Go

This function is available for all modes in the LGR tool. The purpose of this is to provide the U-label/ A-label and codepoints of the input label(s).

Label Forms

Display all label forms

Label

Label can be in U-Label or A-Label form or a list of code points.

Check labels one by one



Click to see the result label form



Display forms

Get label forms on a list of labels

Labels

Choose file No file chosen

File should be a text file encoded in UTF-8 and using 0x0A line ending. It must contain one label per line in U-Label or A-Label form or as a list of code points. Comments begin with '#'

Download forms

Label Forms - Input One Label (Example 1 - valid)

Display all label forms

1. Input is Unicode label

Label

李子

Label can be in U-Label or A-Label form or a list of code points.

2. Click to see the result

Display forms

Code point sequence

U-label

A-label

U+674E U+5B50

李子

xn--i8sx2z

Code point sequence, U-label, A-label - all 3 forms will be generated based on the input

Label Forms - Input One Label (Example 1 - invalid)

Display all label forms

Warning Note

李子。 is invalid as it contains full stop (dot).

1. Input is Unicode label

Label

李子。

Label can be in U-Label or A-Label form or a list of code points.

2. Click to see the result

Display forms

Code point sequence

U-label

A-label

U+674E U+5B50 U+3002

-

-

Code point sequence will be generated,
U-label, A-label forms will not be generated.

Label Forms - Input One Label (Example 2 - valid)

Display all label forms

1. Input is Code point sequence

Label

U+101D U+1031 U+1006 U+102C

Label can be in U-Label or A-Label form or a list of code points.

2. Click to see the result

Display forms

Code point sequence	U-label	A-label
U+101D U+1031 U+1006 U+102C	ᄁᄂᄃᄄ	xn--tid9b4bs

Code point sequence, U-label, A-label - all 3 forms will be generated based on the input

Label Forms - Input One Label (Example 2 - invalid)

Display all label forms

Warning Note

ᄀᄀᄀ. is invalid as it contains full stop (dot).

1. Input is Unicode label

Label

U+101D U+1031 U+1006 U+102C U+3002

Label can be in U-Label or A-Label form or a list of code points.

2. Click to see the result

Display forms

Code point sequence	U-label	A-label
U+101D U+1031 U+1006 U+102C U+3002	-	-

Code point sequence will be generated,
U-label, A-label forms will not be generated.

Label Forms - Input a List of Labels (Example 3)

Upload file to check the list of labels
(in txt format)

Get label forms on a list of labels

Labels

Choose file No file chosen

File should be a text file encoded in UTF-8 and using 0x0A line ending. It must contain one label per line in U-Label or A-Label form or as a list of code points. Comments begin with '#'

Download forms

Download the result file of Label forms
(in csv format)

Label Forms - Input a List of Labels (Example 3 - Input)



example3.txt

Downloads/example3.txt ↕

```
#Code points - valid
U+1005 U+102D U+1014 U+103A
#Code points - invalid
U+1005 U+102D U+1014 U+103A U+002D
#Unicode - valid
ရတနာ
#Unicode - invalid
ရတ--နာ
#Punycode - valid
xn--ridd1ji0d
#Punycode - invalid
xn--vi8hua2f
```

- Input file format is .txt.
- Input can be one label per line.
- Each label can be in
 - Code point sequence
 - Unicode
 - Punycode
- The lines beginning with '#' are comments and will not be processed.

Label Forms - Input a List of Labels (Example 3 - Output)

Input	Code point sequence	U-label	A-label	Note
U+1005 U+102D U+1014 U+103A	U+1005 U+102D U+1014 U+103A	စီနီ	xn--sid4a9d7b	-
U+1005 U+102D U+1014 U+103A U+002D	U+1005 U+102D U+1014 U+103A U+002D	-	-	စီနီ- is invalid due to hyphen restrictions in the RFC5891 as it ends with a hyphen-minus.
ရတနာ	U+101B U+1010 U+1014 U+102C	ရတနာ	xn--3idit4e	-
ရတ--နာ	U+101B U+1010 U+002D U+002D U+1014 U+102C	-	-	ရတ--နာ is invalid due to hyphen restrictions in the RFC5891 as it contains hyphen-minus in the third and fourth positions.
xn--ridd1ji0d	U+1006 U+102D U+102F U+1004 U+103A	ဆိုင်	xn--ridd1ji0d	-
xn--vi8hua2f	U+D83C U+DF55 U+D83C U+DF5F U+D83C U+DF79	-	-	'utf-8' codec can't encode characters in position 55-60: surrogates not allowed

- Output csv: *U-label* and *A-label* forms will be generated when
 - Input is Code point sequence and valid
 - Input is U-label and valid
 - Input is A-label and valid

Label Forms - Input a List of Labels (Example 3 - Output)

Input	Code point sequence	U-label	A-label	Note
U+1005 U+102D U+1014 U+103A	U+1005 U+102D U+1014 U+103A	စီနီ	xn--sid4a9d7b	-
U+1005 U+102D U+1014 U+103A U+002D	U+1005 U+102D U+1014 U+103A U+002D	-	-	စီနီ- is invalid due to hyphen restrictions in the RFC5891 as it ends with a hyphen-minus.
ရတနာ	U+101B U+1010 U+1014 U+102C	ရတနာ	xn--3idit4e	-
ရတ--နာ	U+101B U+1010 U+002D U+002D U+1014 U+102C	-	-	ရတ--နာ is invalid due to hyphen restrictions in the RFC5891 as it contains hyphen-minus in the third and fourth positions.
xn--ridd1ji0d	U+1006 U+102D U+102F U+1004 U+103A	ဆိုင်	xn--ridd1ji0d	-
xn--vi8hua2f	U+D83C U+DF55 U+D83C U+DF5F U+D83C U+DF79	-	-	'utf-8' codec can't encode characters in position 55-60: surrogates not allowed

- Output csv: Notes will be generated, (with no A-label or U-label) when
 - Input is Code point sequence and IDNA2008-non-compliant
 - Input is U-label and IDNA2008-non-compliant
 - Input is A-label and IDNA2008-non-compliant

Engage with ICANN and IDN Program



Thank You and Questions

Reach us at: IDNProgram@icann.org

Website: icann.org/idn



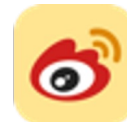
twitter.com/icann



[gplus.to/icann](https://plus.google.com/icann)



facebook.com/icannorg



weibo.com/ICANNorg



linkedin.com/company/icann



flickr.com/photos/icann



youtube.com/user/icannnews



slideshare.net/icannpresentations