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

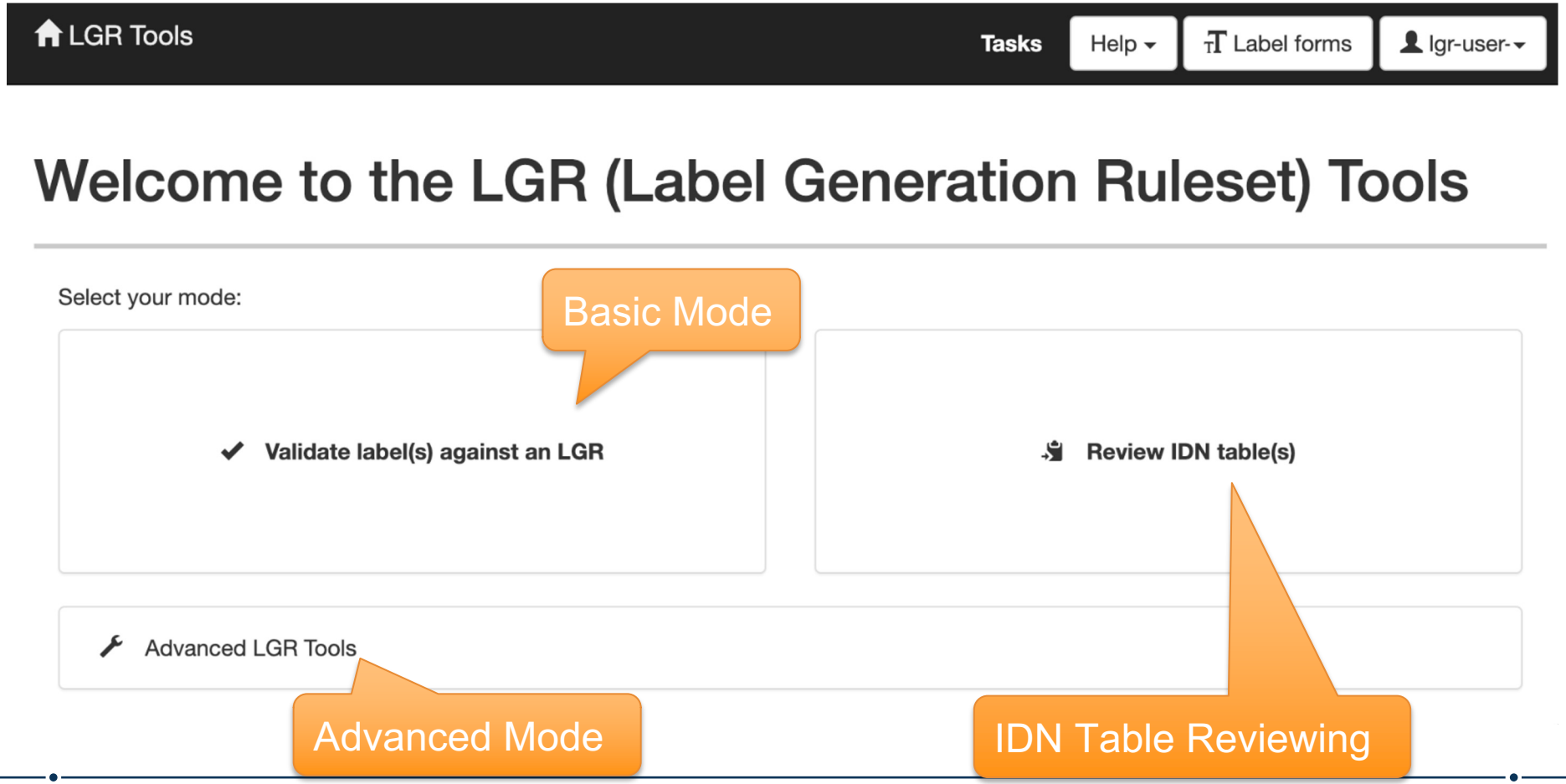
Agenda Details (Cont'd)

- **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**



Advanced LGR Tools

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

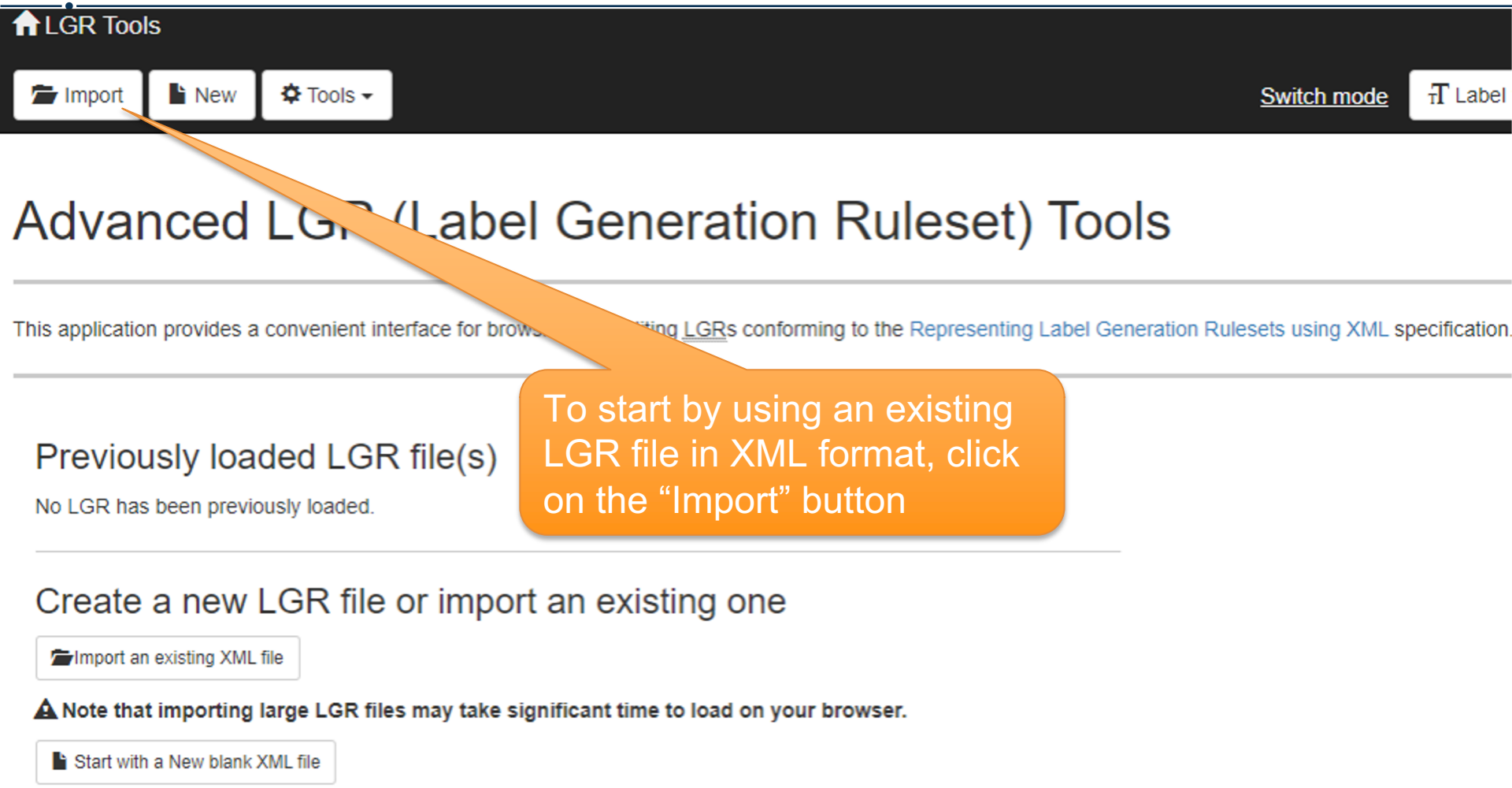
The following files contains your tools computation results.

⚠ Note that these files could be cleaned up regularly.

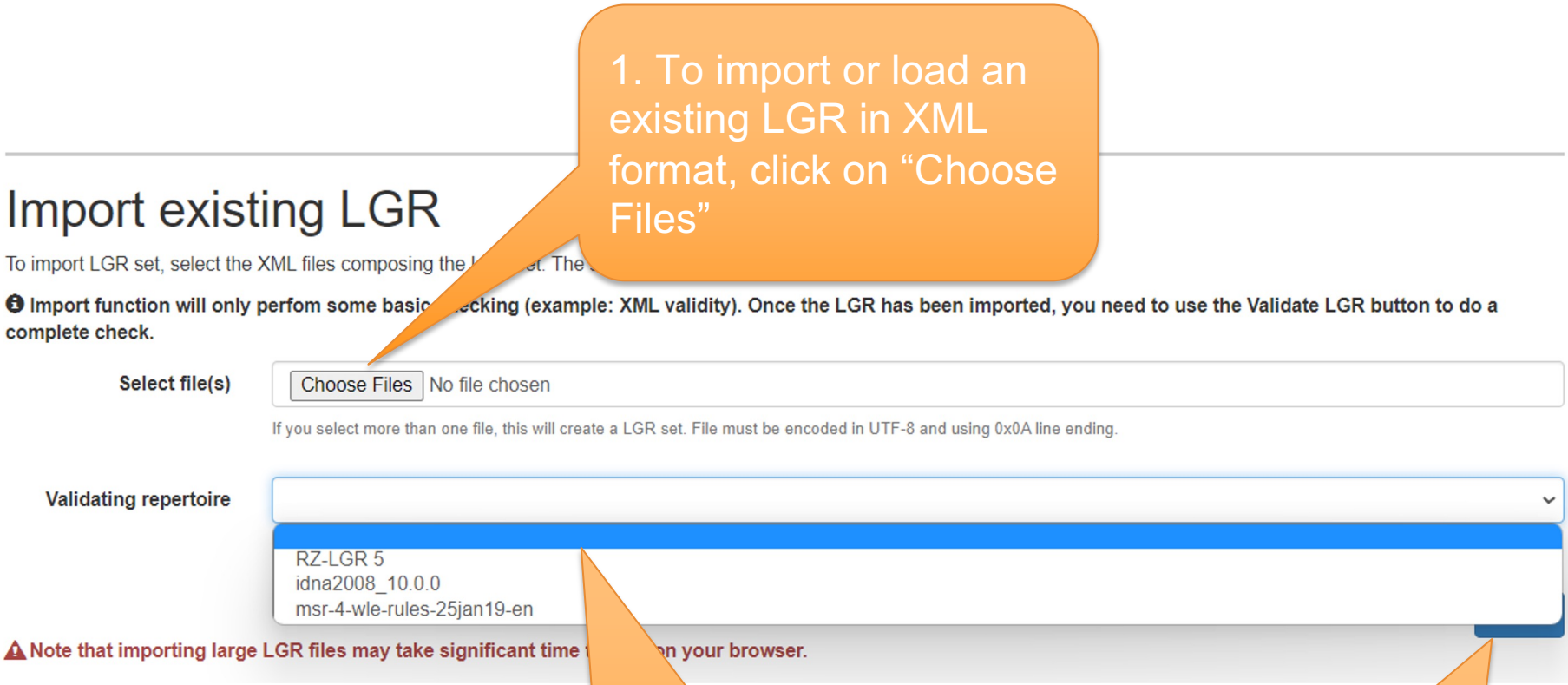
- Download 20221022_171249_annotation_RZ-LGR_5.txt.gz

Import or Load LGR

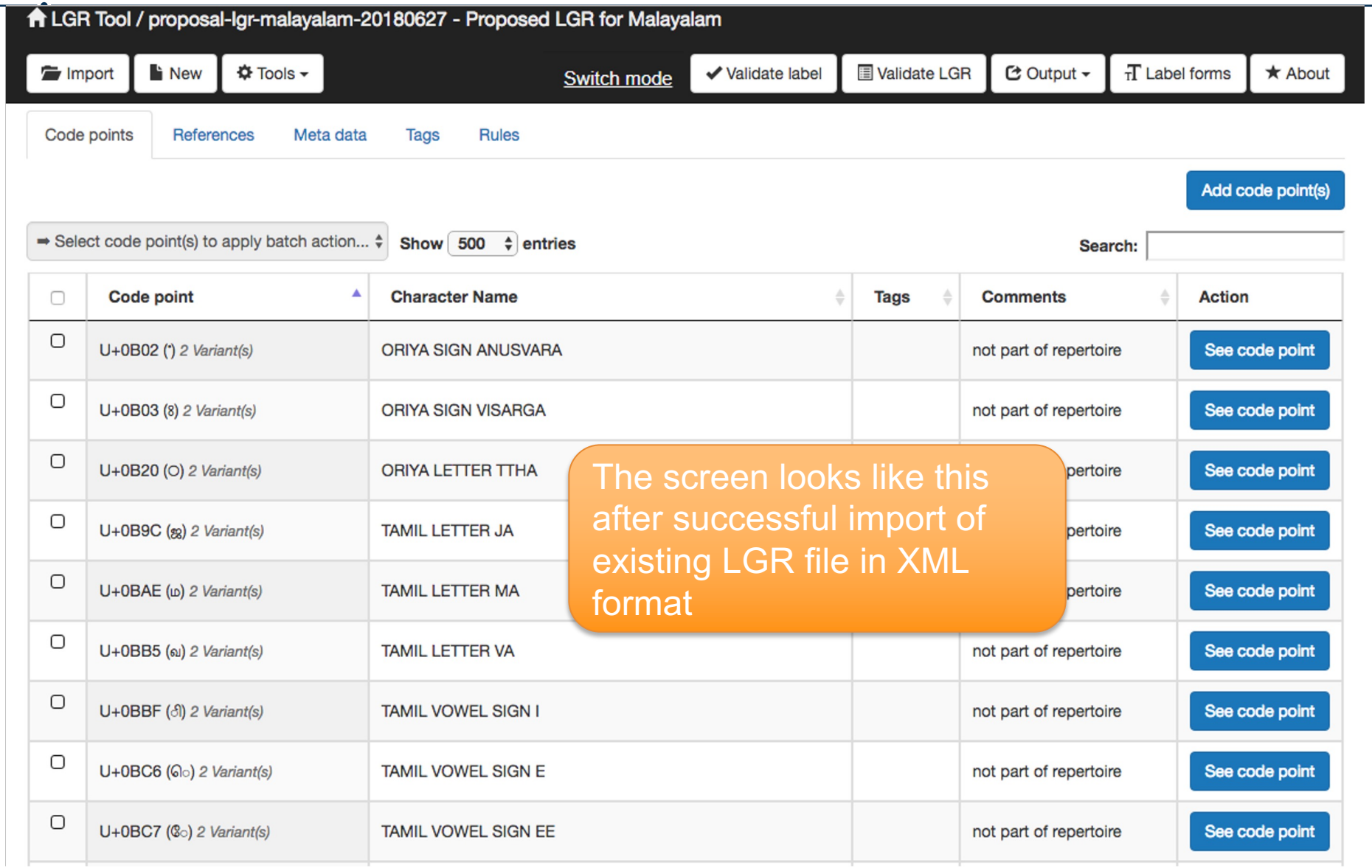
Import or Load LGR



Select LGR with Validating Repertoire

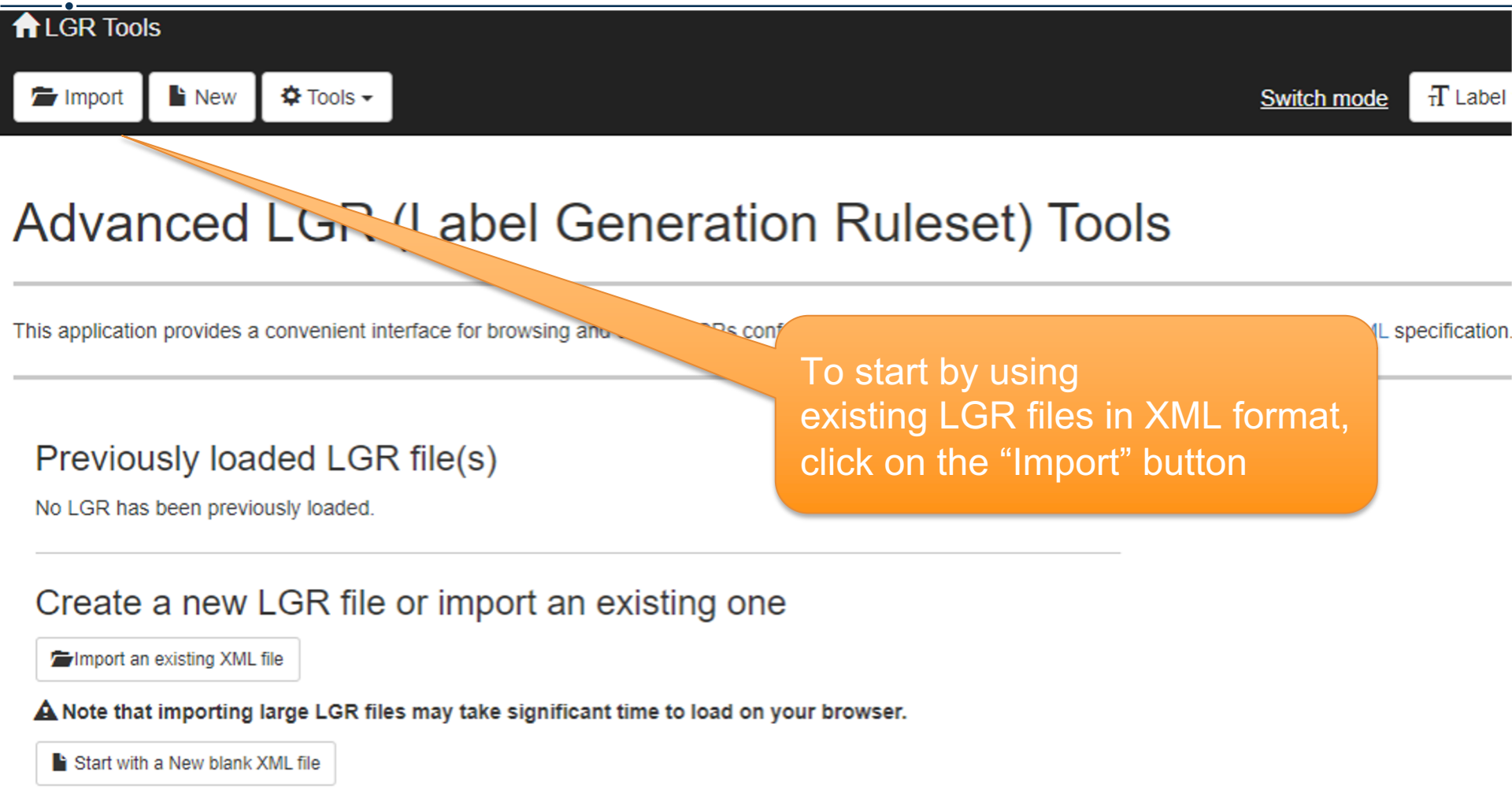


Imported LGR

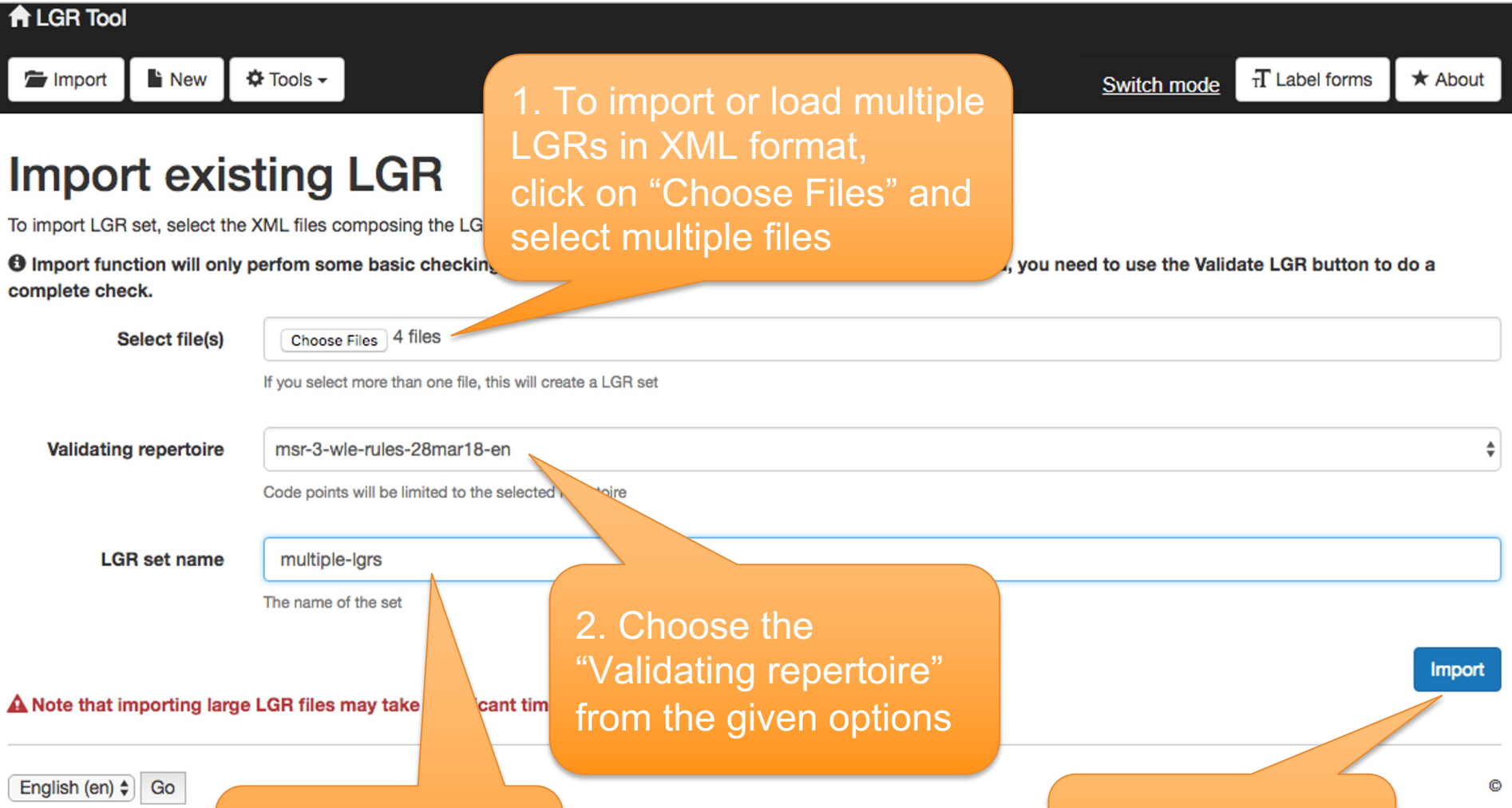


Import Multiple LGRs

Import or Load LGRs



Select LGRs with Validating Repertoire



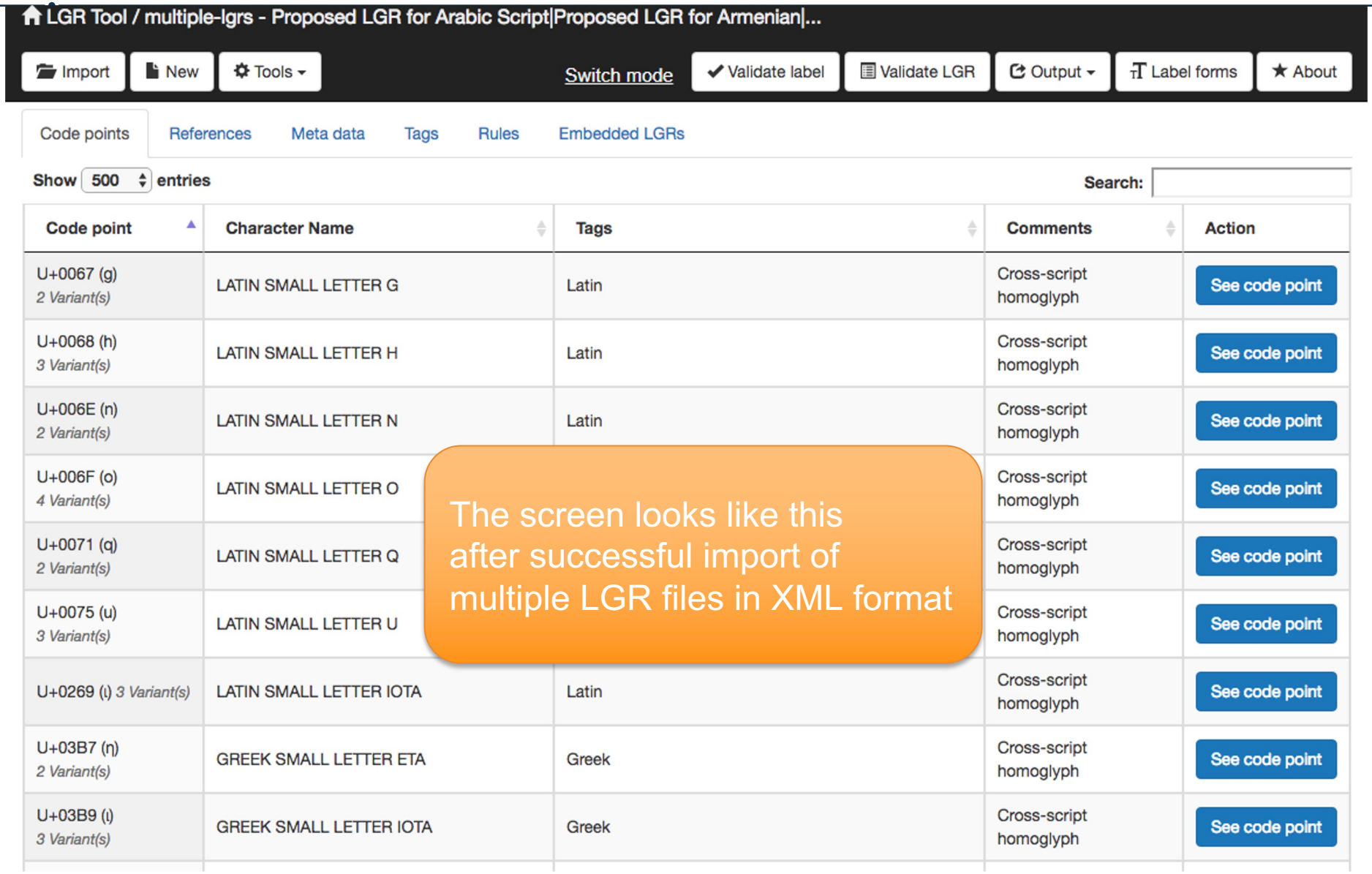
1. To import or load multiple LGRs in XML format, click on “Choose Files” and select multiple files

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

3. Enter a name for the set of LGRs

4. Click on the “Import” button

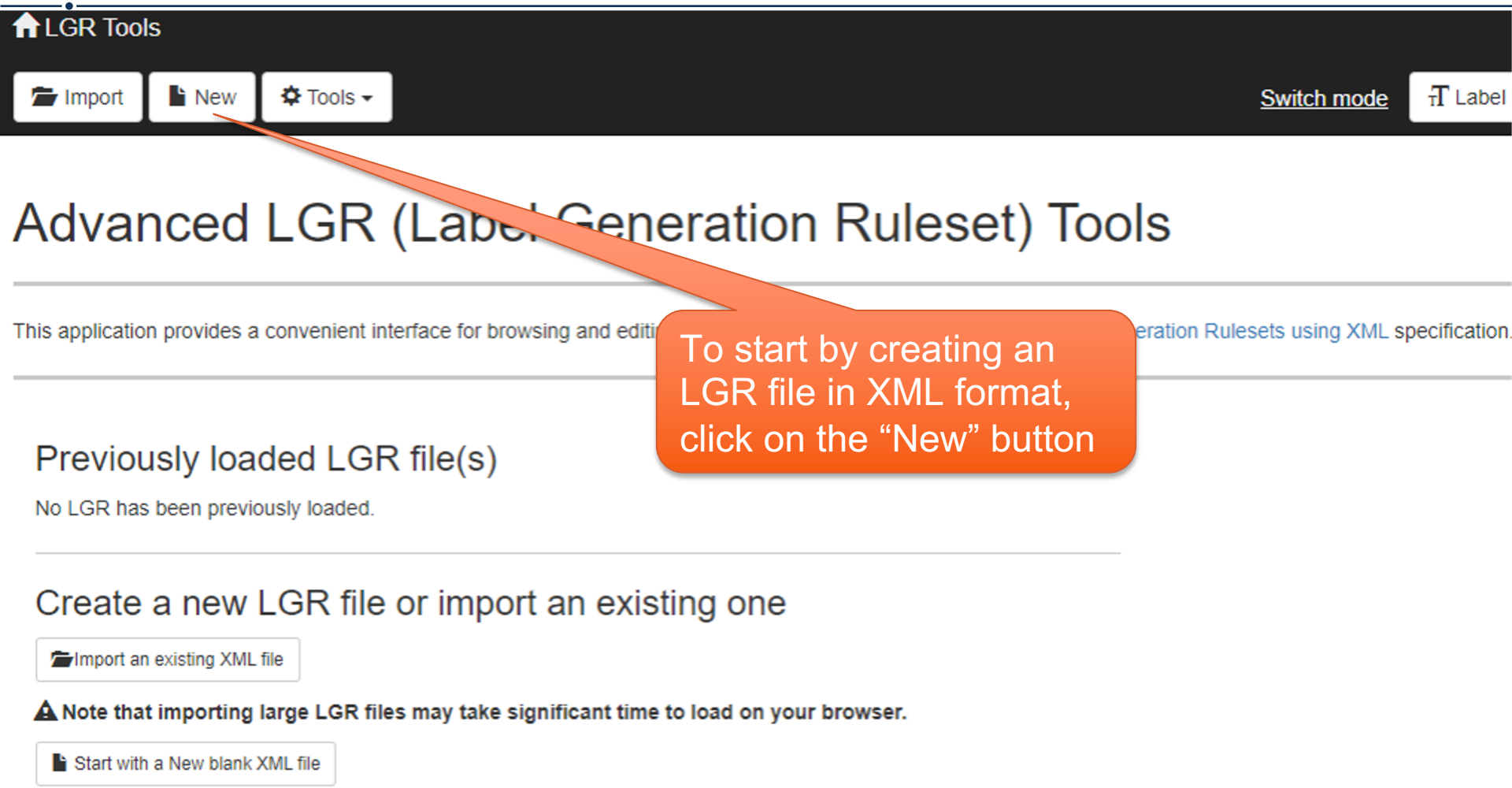
Imported LGRs As a Merged LGR



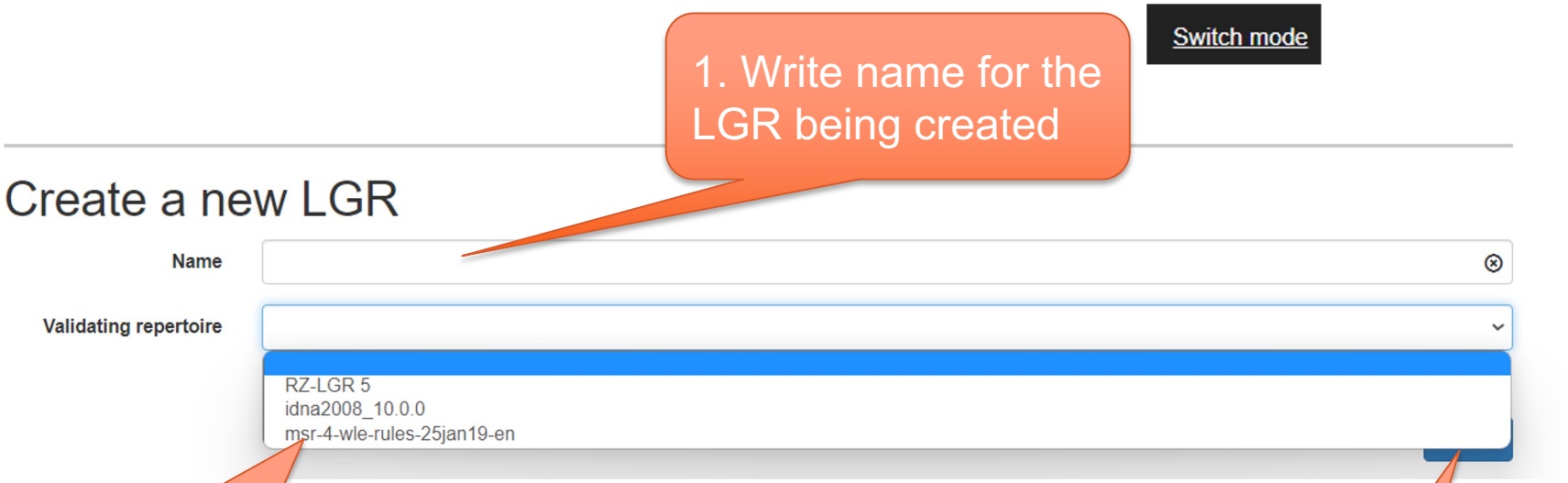
The screen looks like this after successful import of multiple LGR files in XML format

Create LGR

Create an LGR



Enter LGR Details

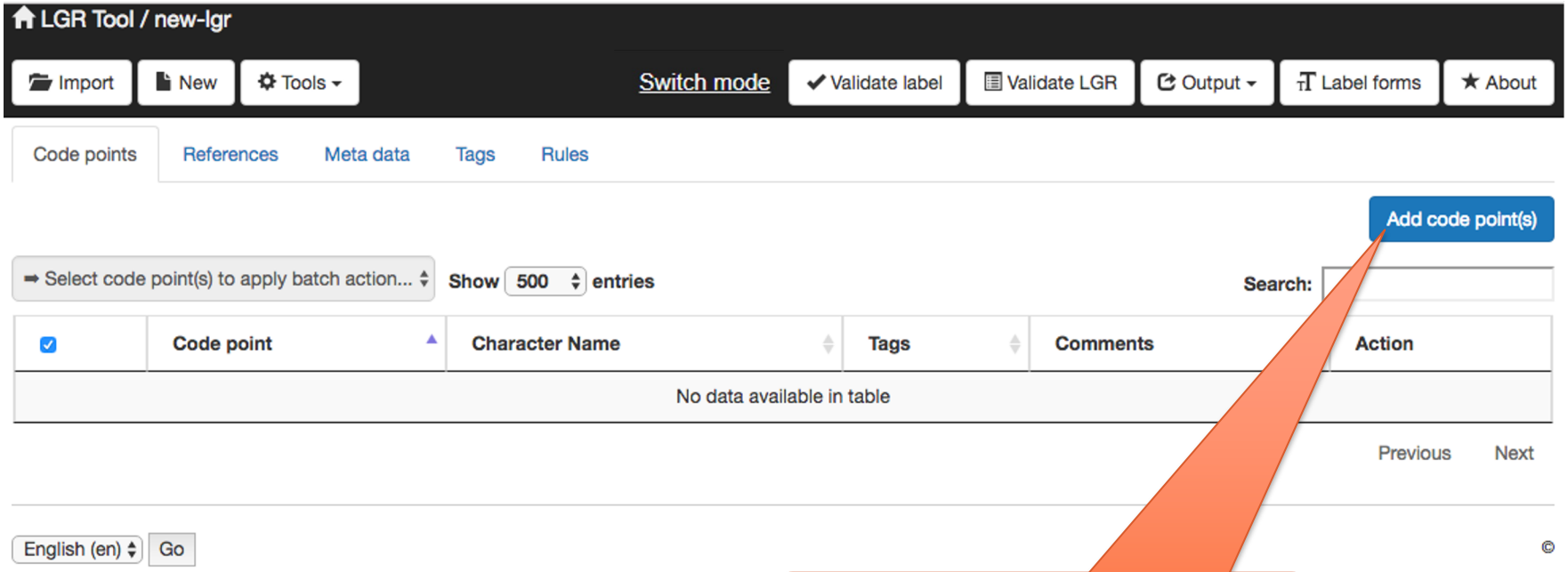


1. Write name for the LGR being created

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

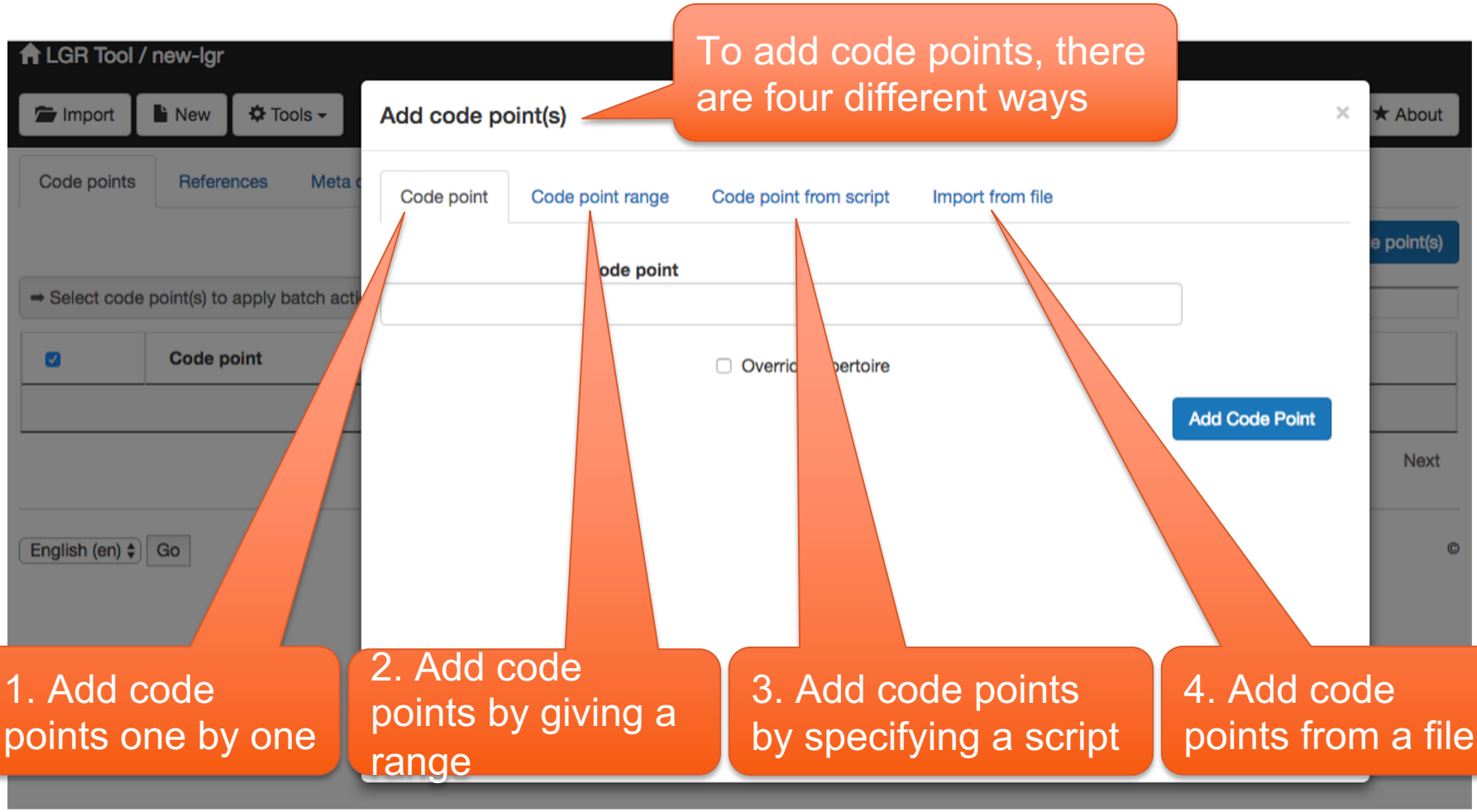
3. Click on the “Create” button

Add Code Points

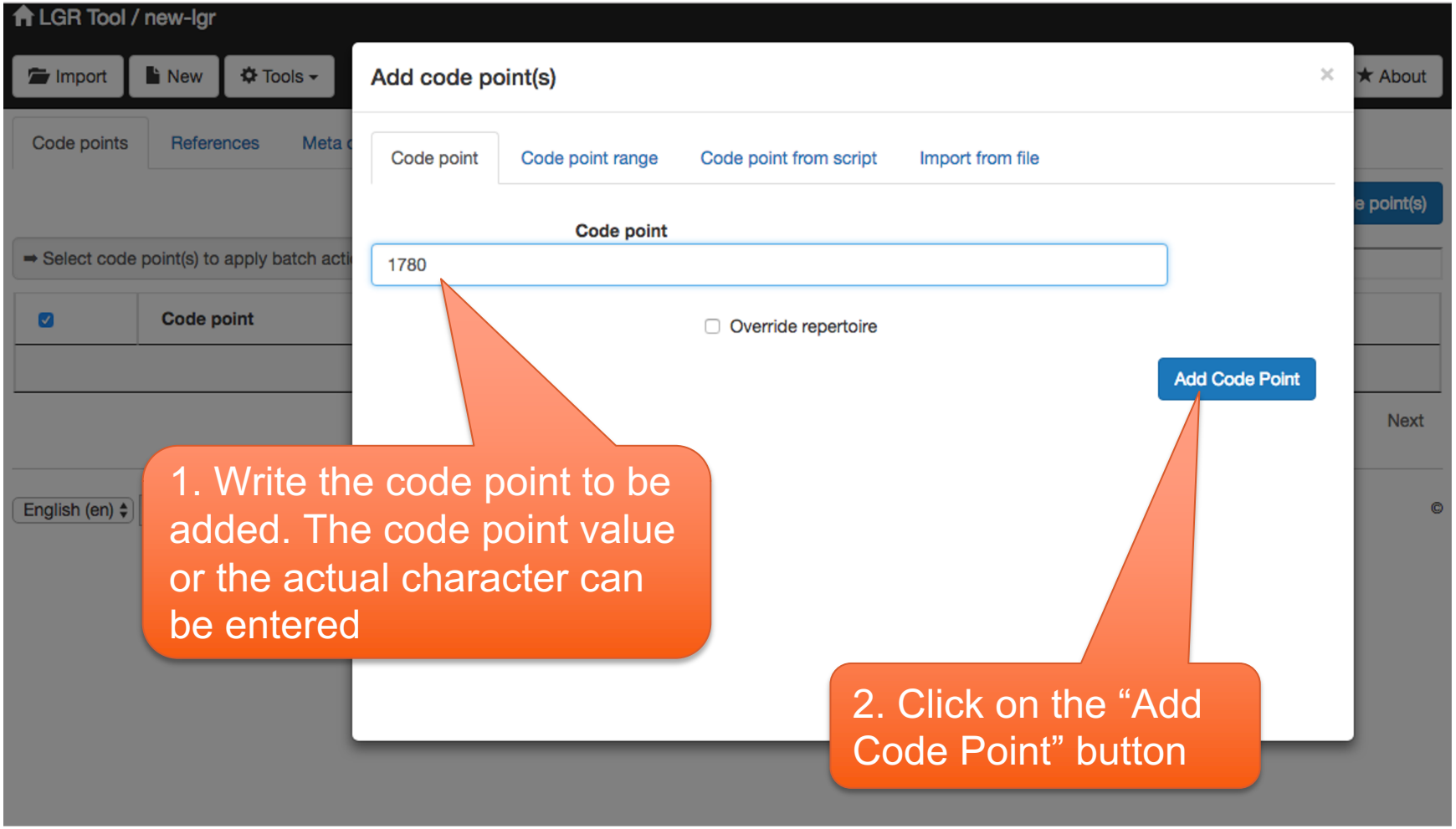


Click on “Add code points” to add code points to the newly created LGR

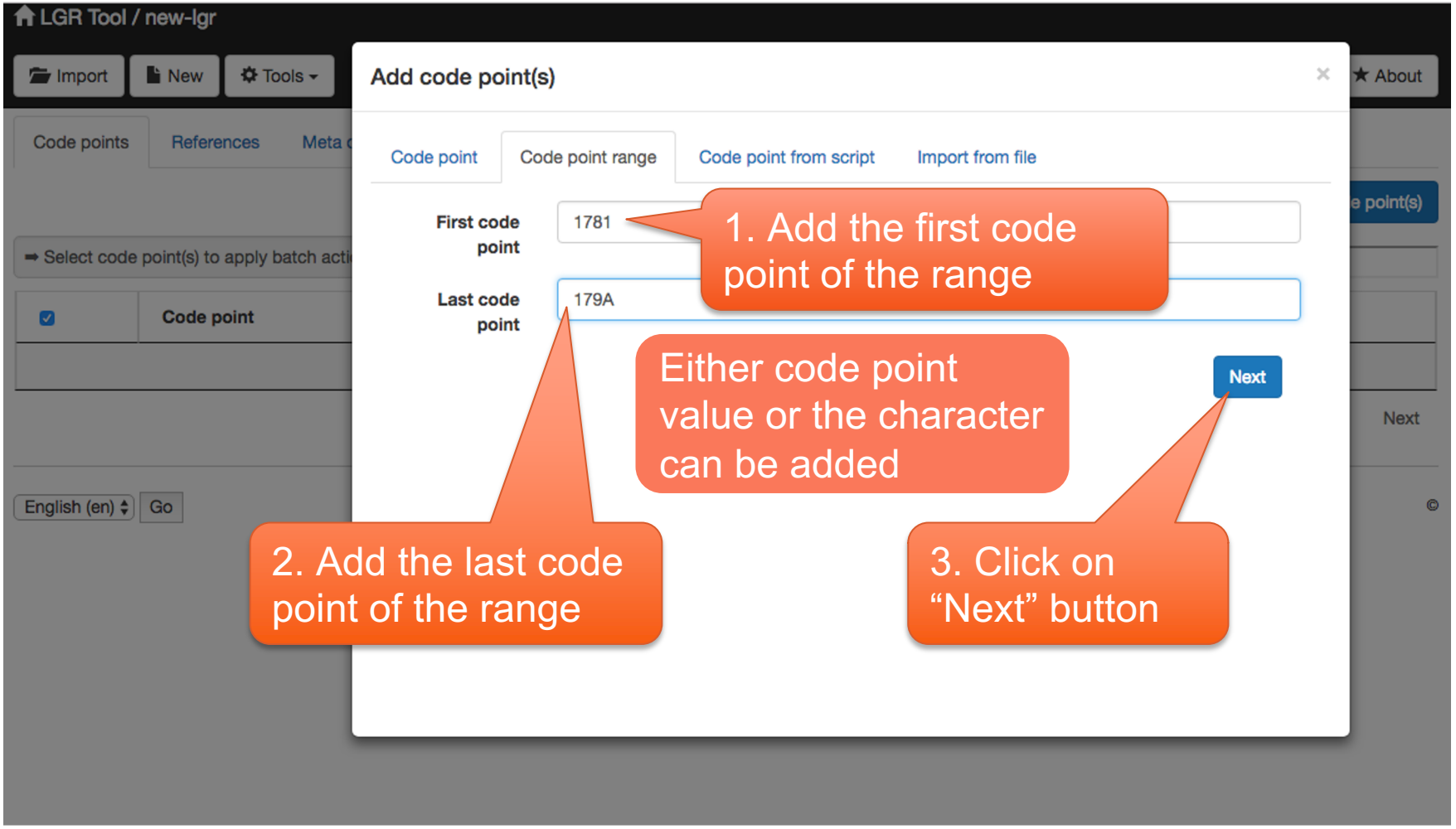
Add Code Points



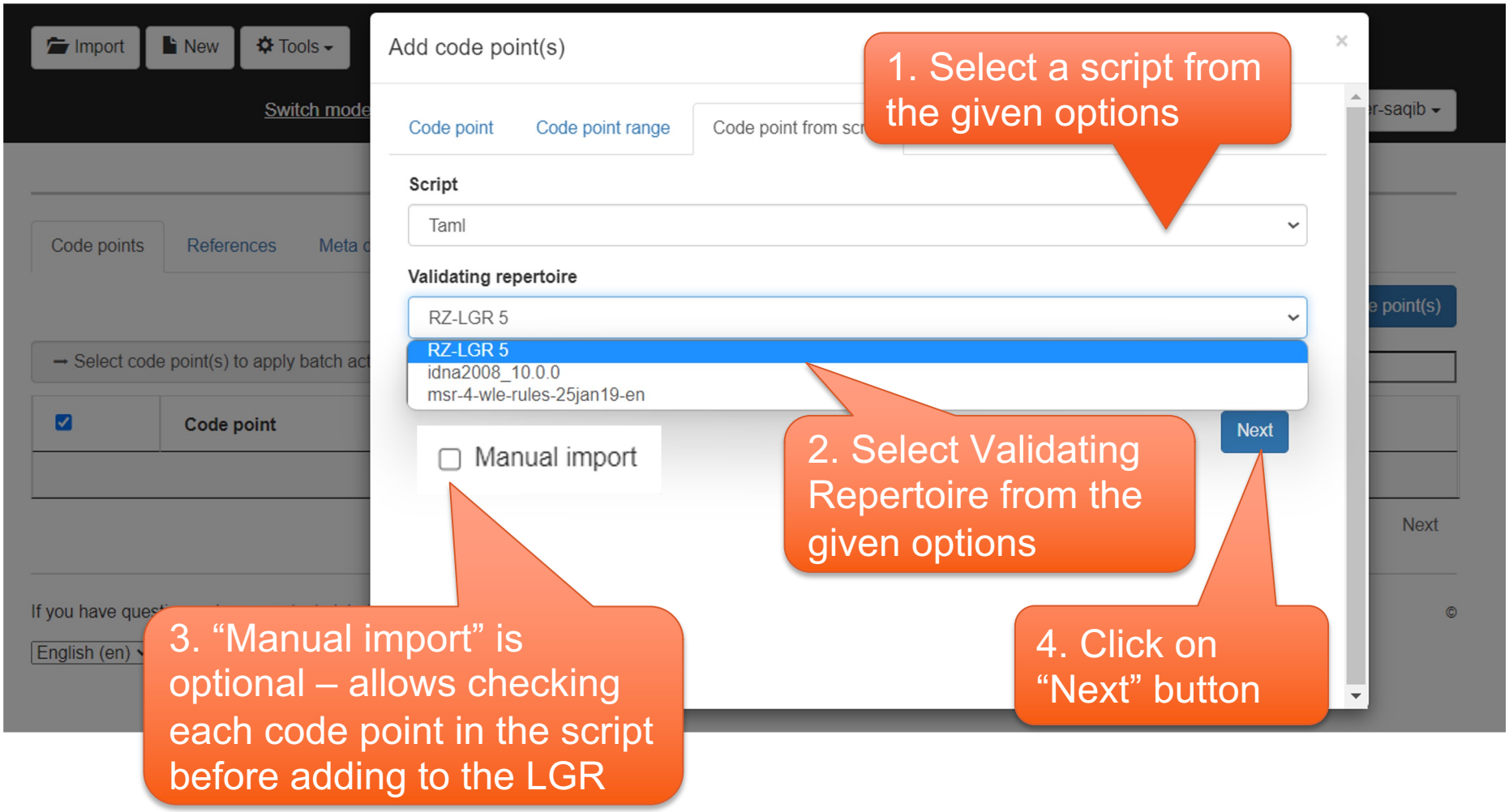
Add Code Points



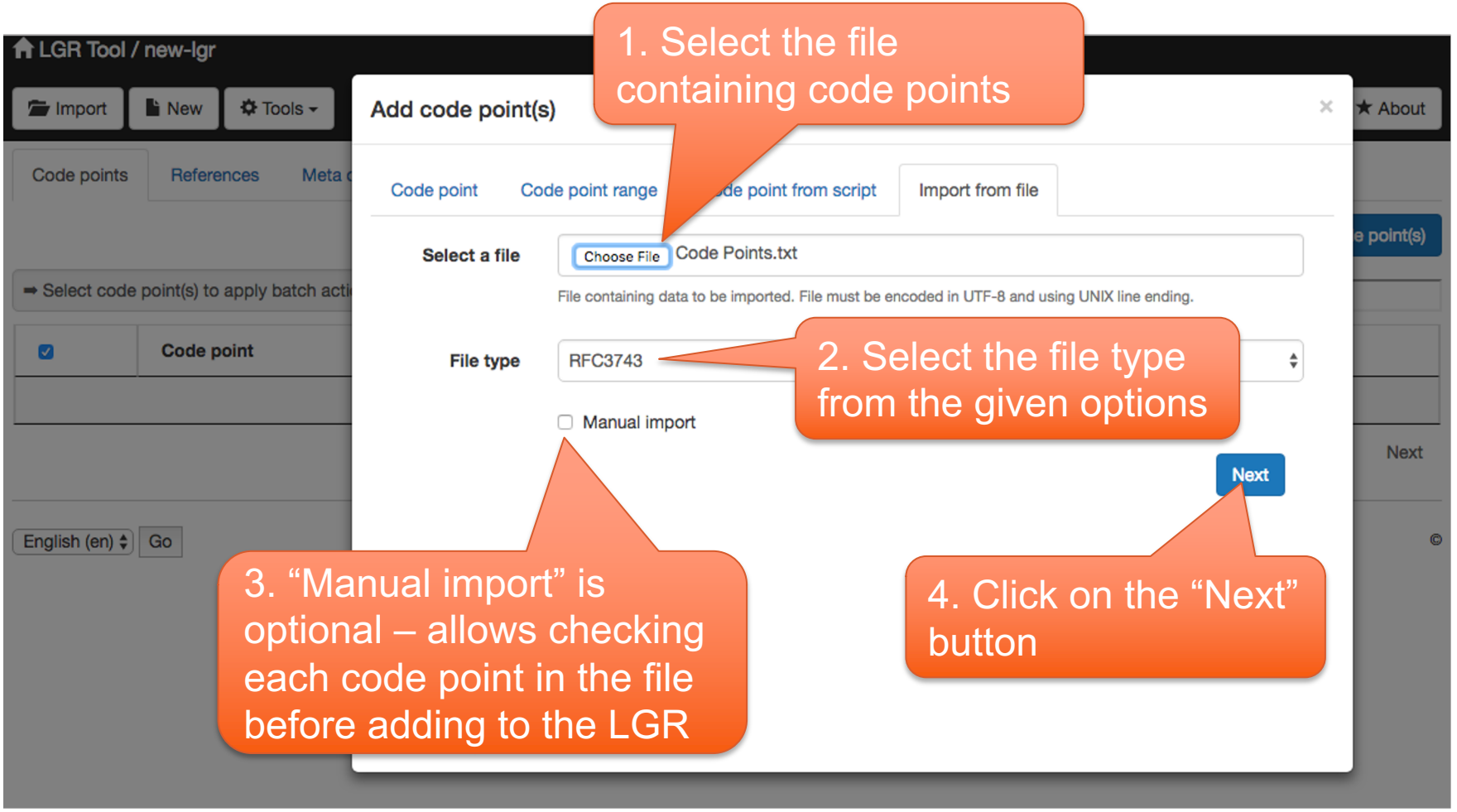
Add Code Points



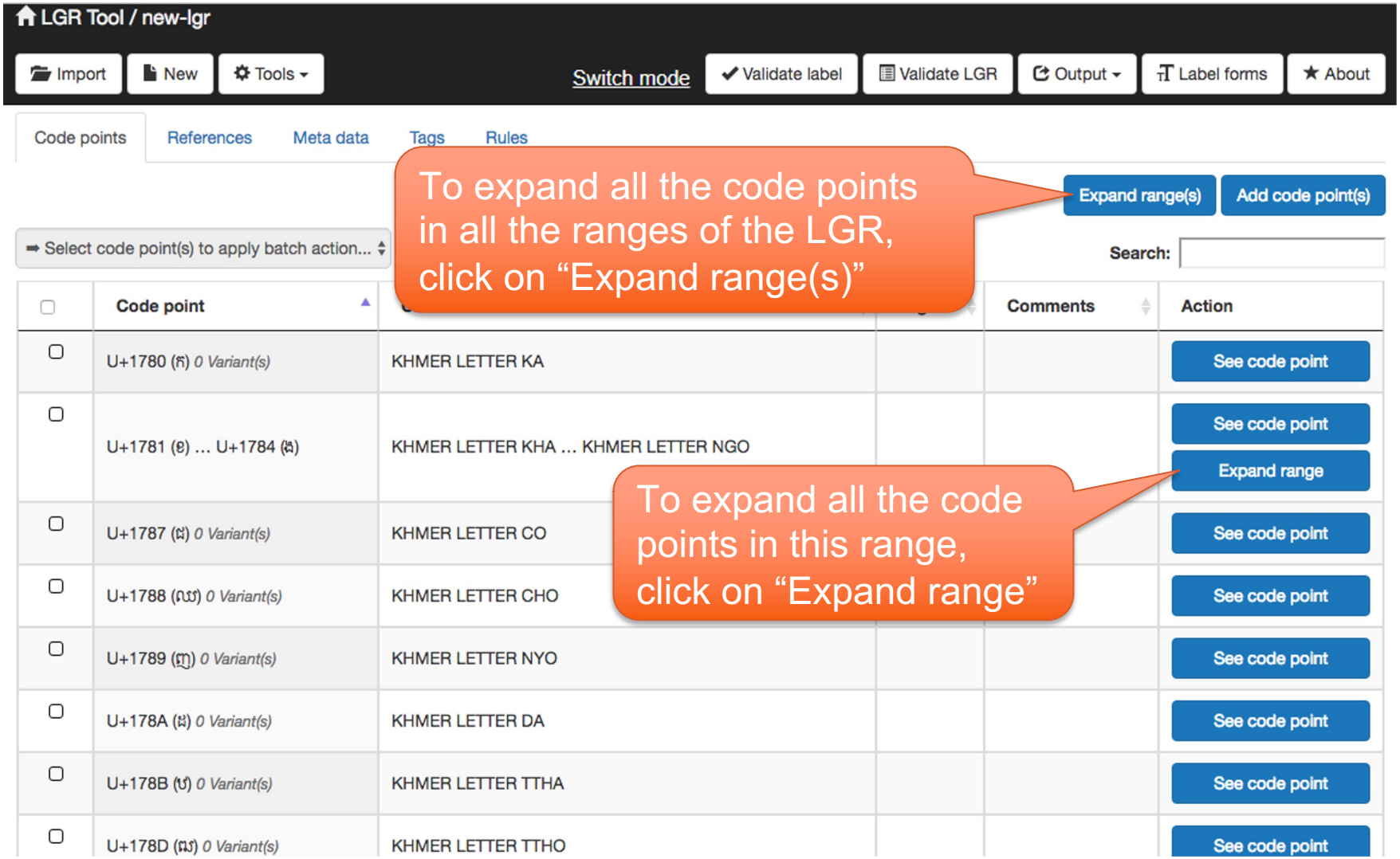
Add Code Points



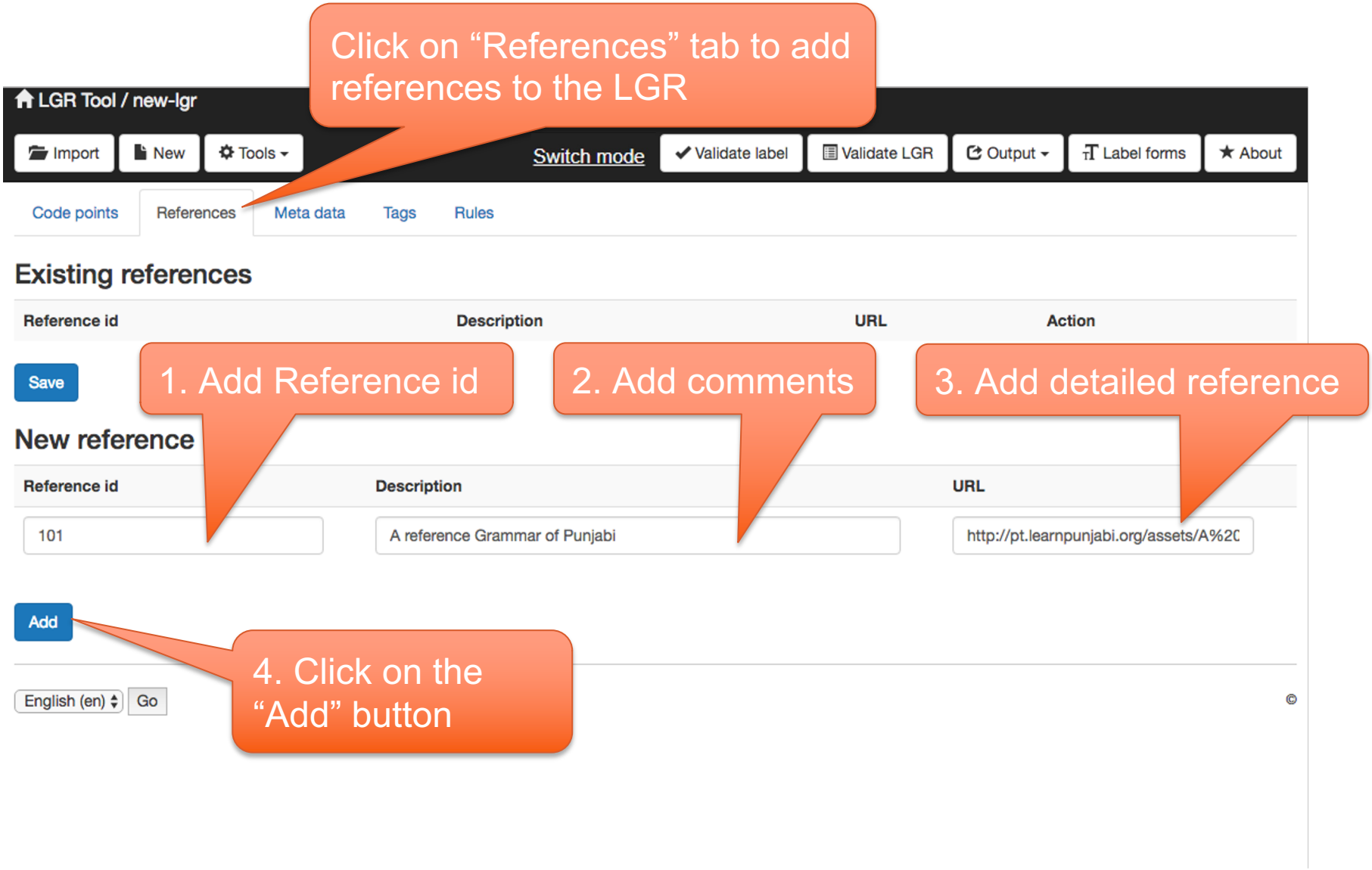
Add Code Points



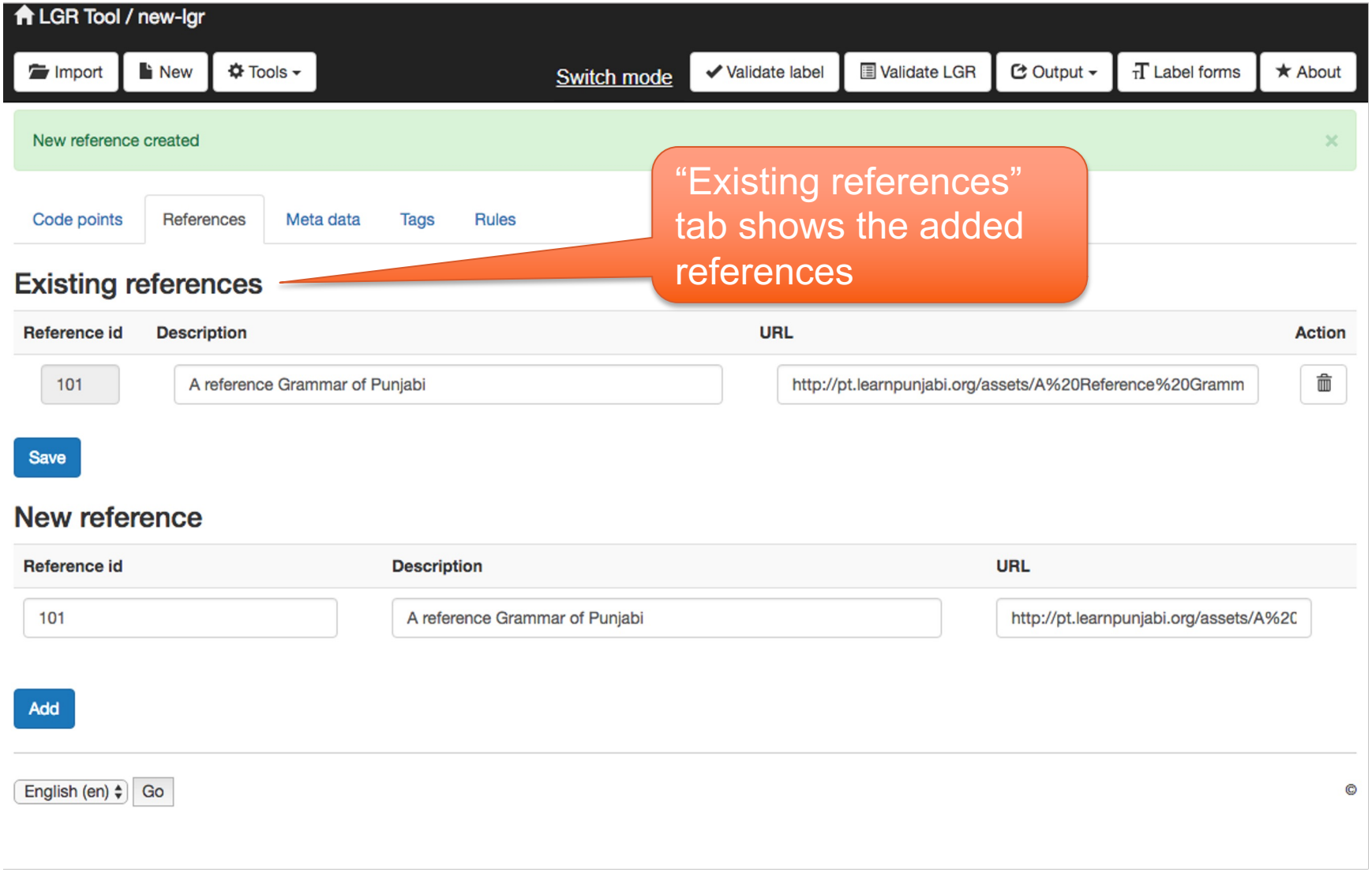
Expand Ranges



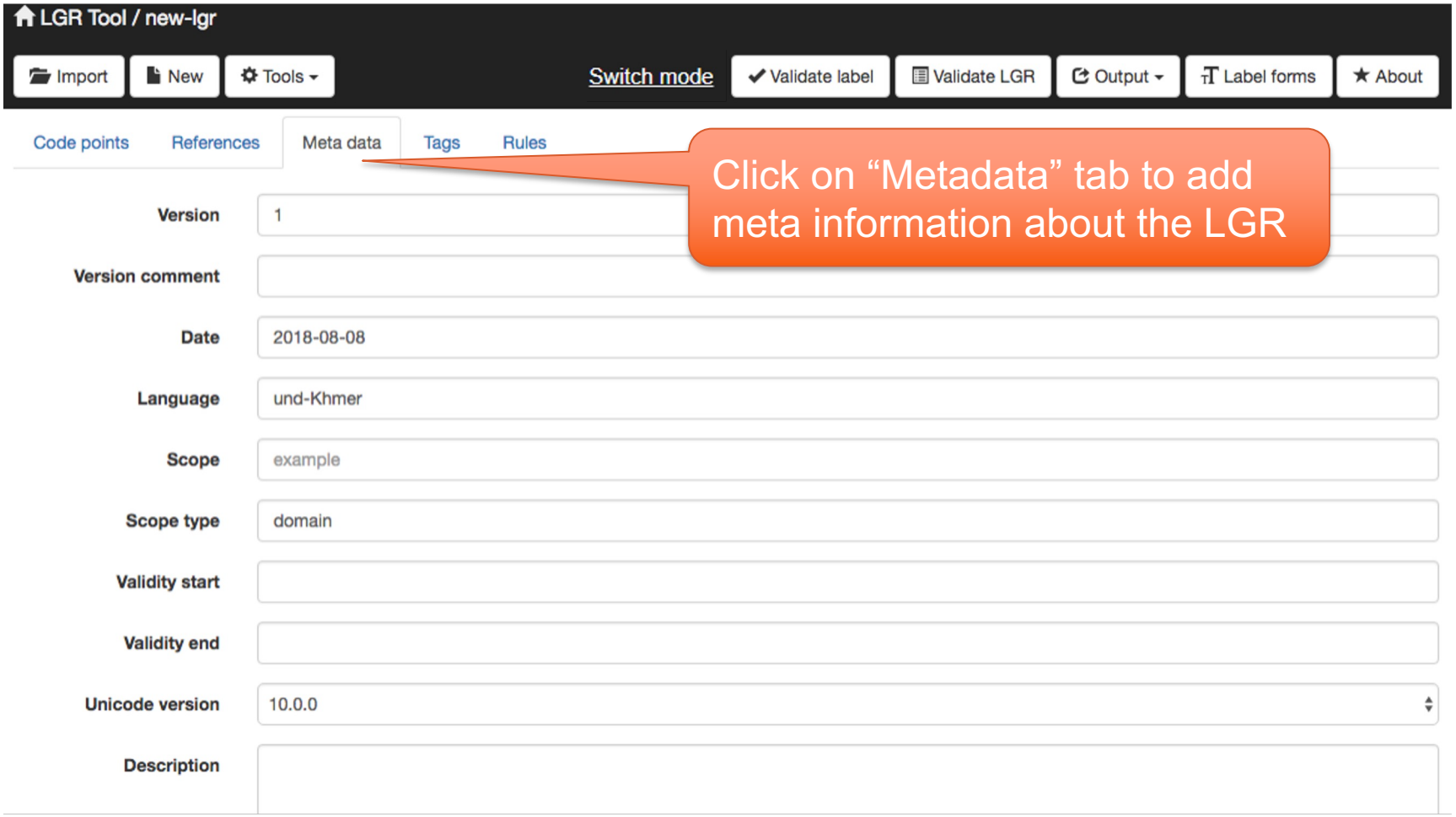
Add References



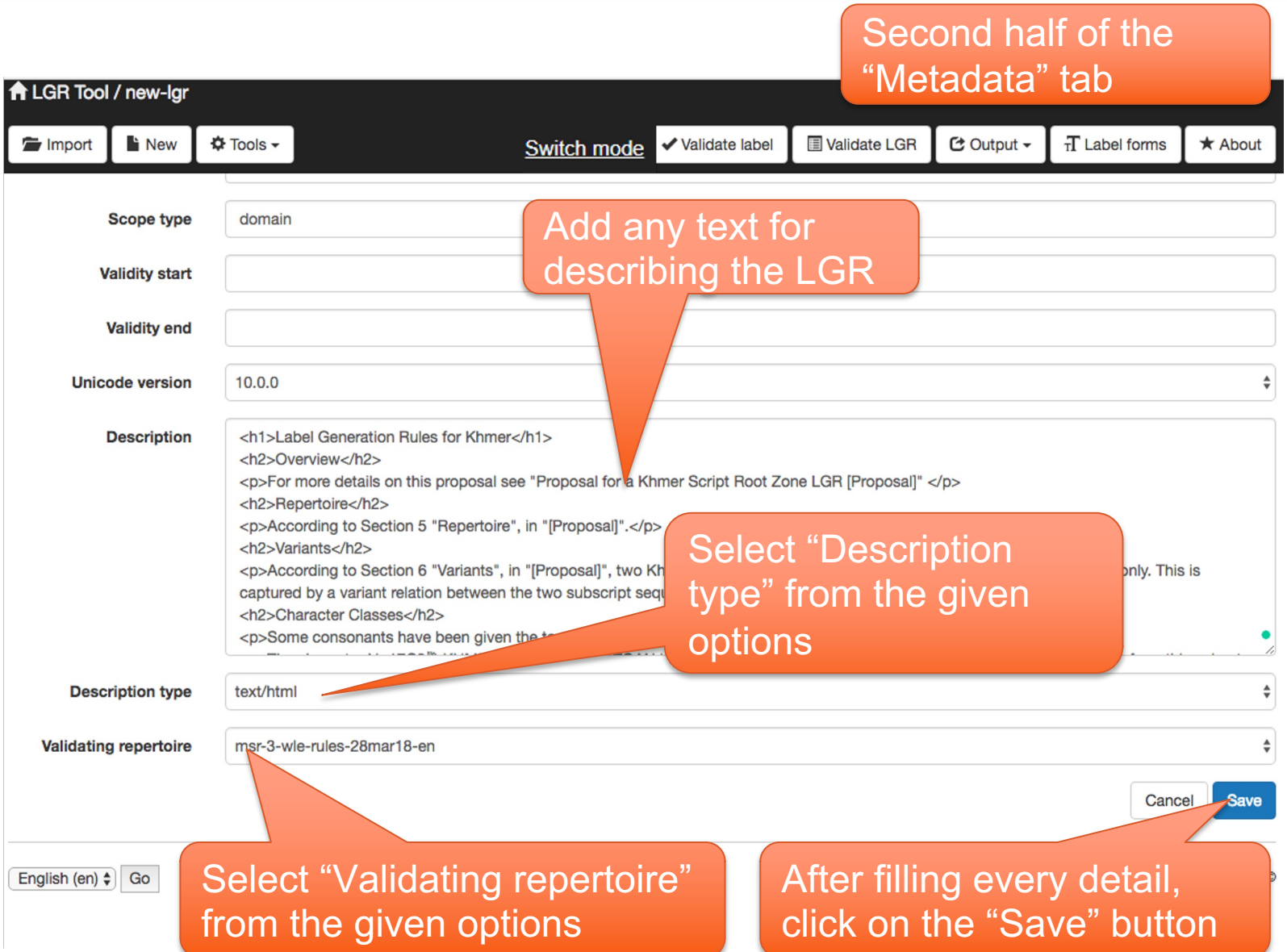
Add References



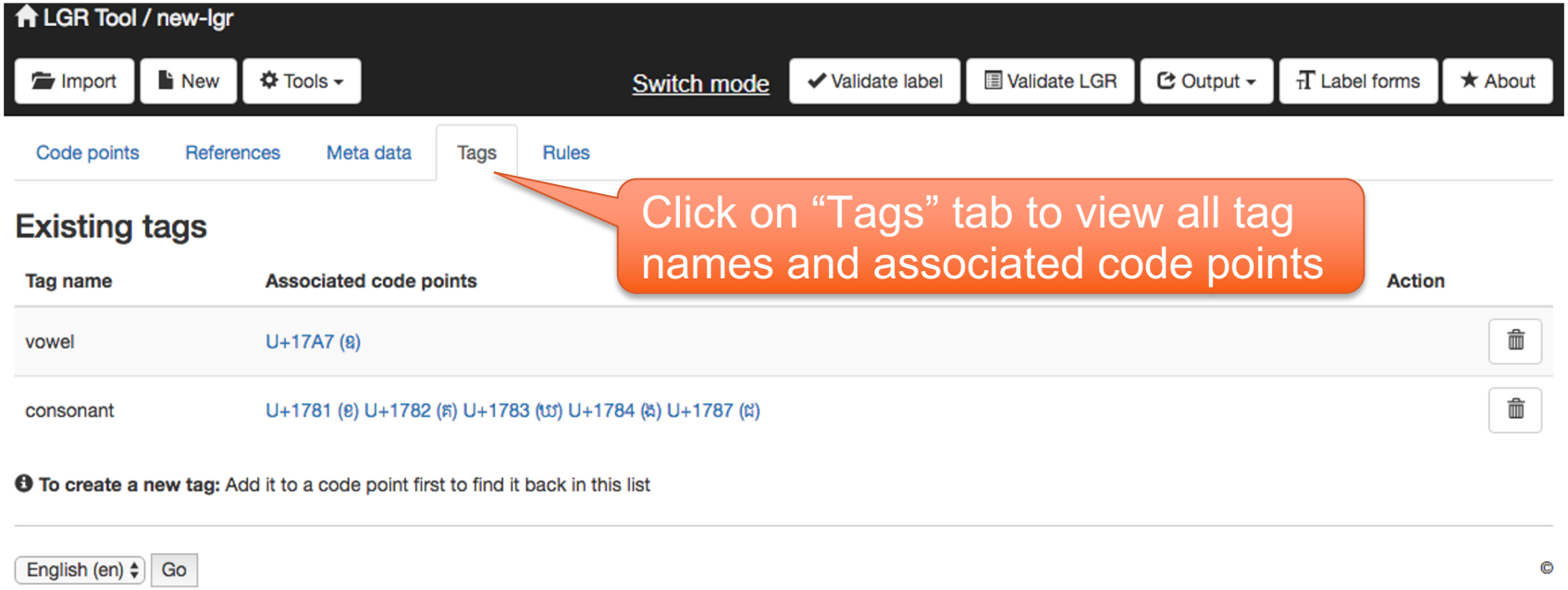
Define Metadata



Define Meta Data

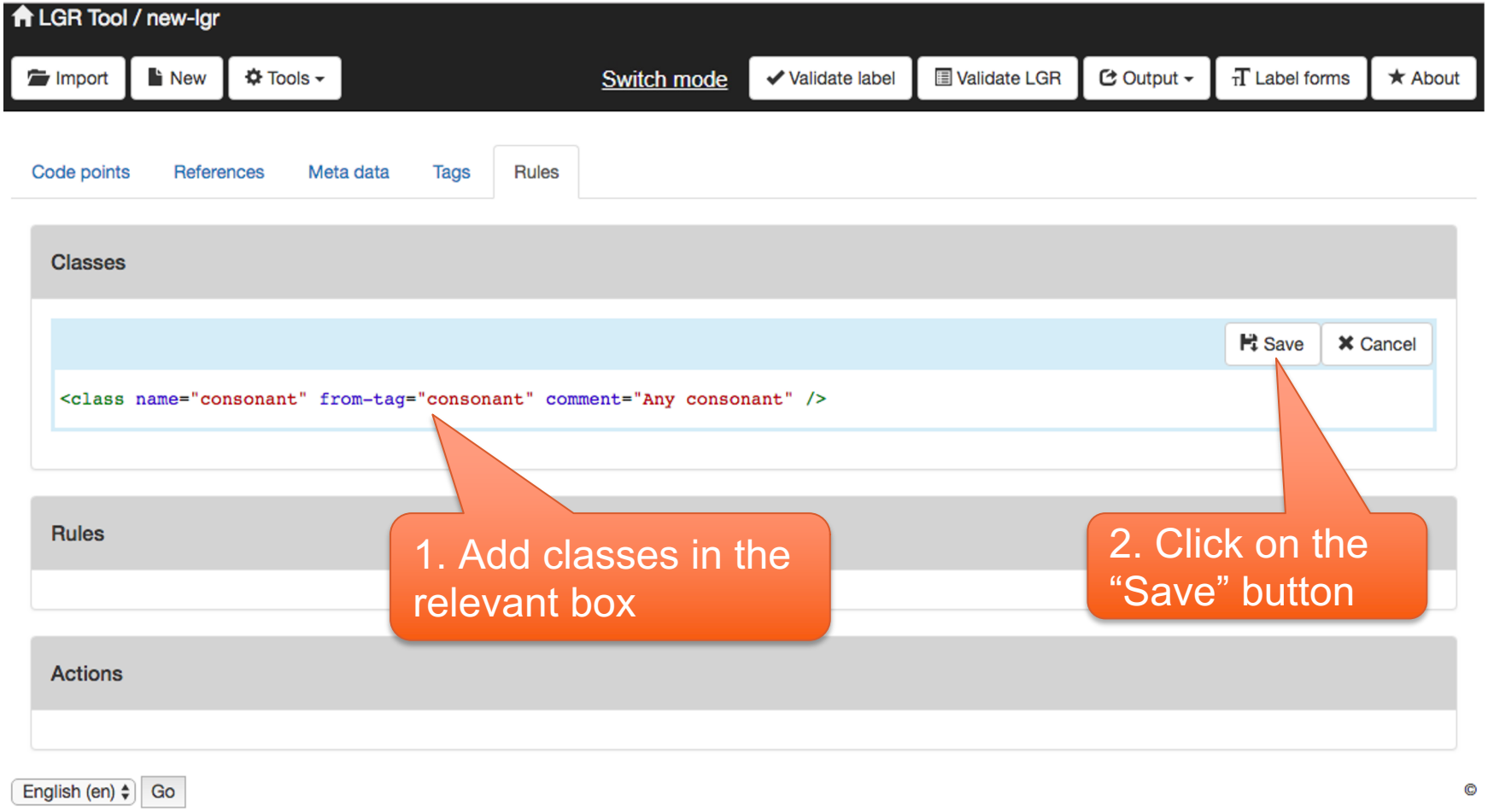


View Tags

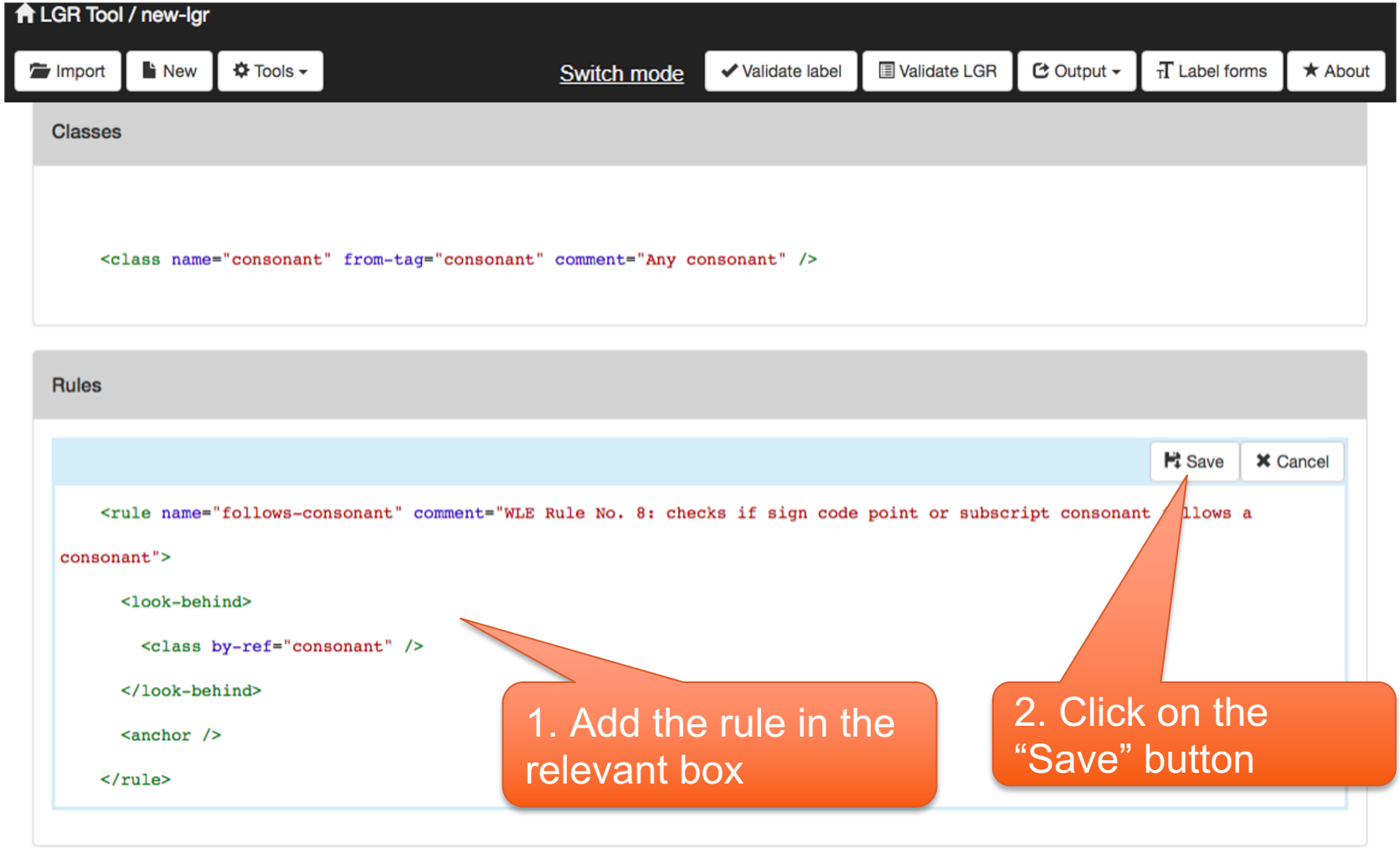

LGR Tool / new-lgr
Import
New
Tools
Switch mode
Validate label
Validate LGR
Output
Label forms
About
Code points
References
Meta data
Tags
Rules
Click on “Tags” tab to view all tag names and associated code points
Existing tags
Tag name
Associated code points
Action
vowel
U+17A7 (ꢧ)
consonant
U+1781 (ꢁ) U+1782 (ꢂ) U+1783 (ꢃ) U+1784 (ꢄ) U+1787 (ꢇ)
To create a new tag: Add it to a code point first to find it back in this list
English (en)
Go
©

Define Classes, Rules & Actions

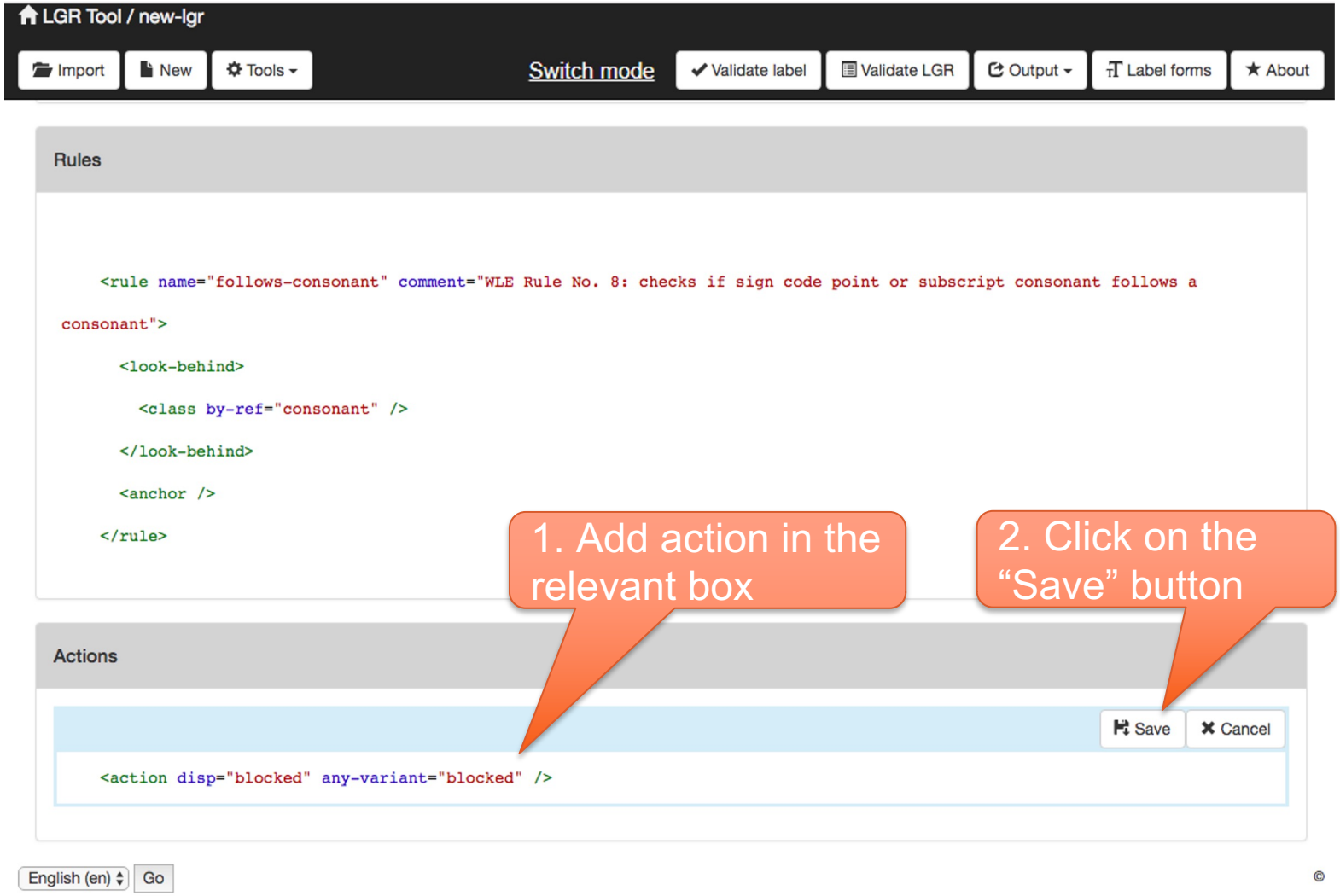
Define Classes



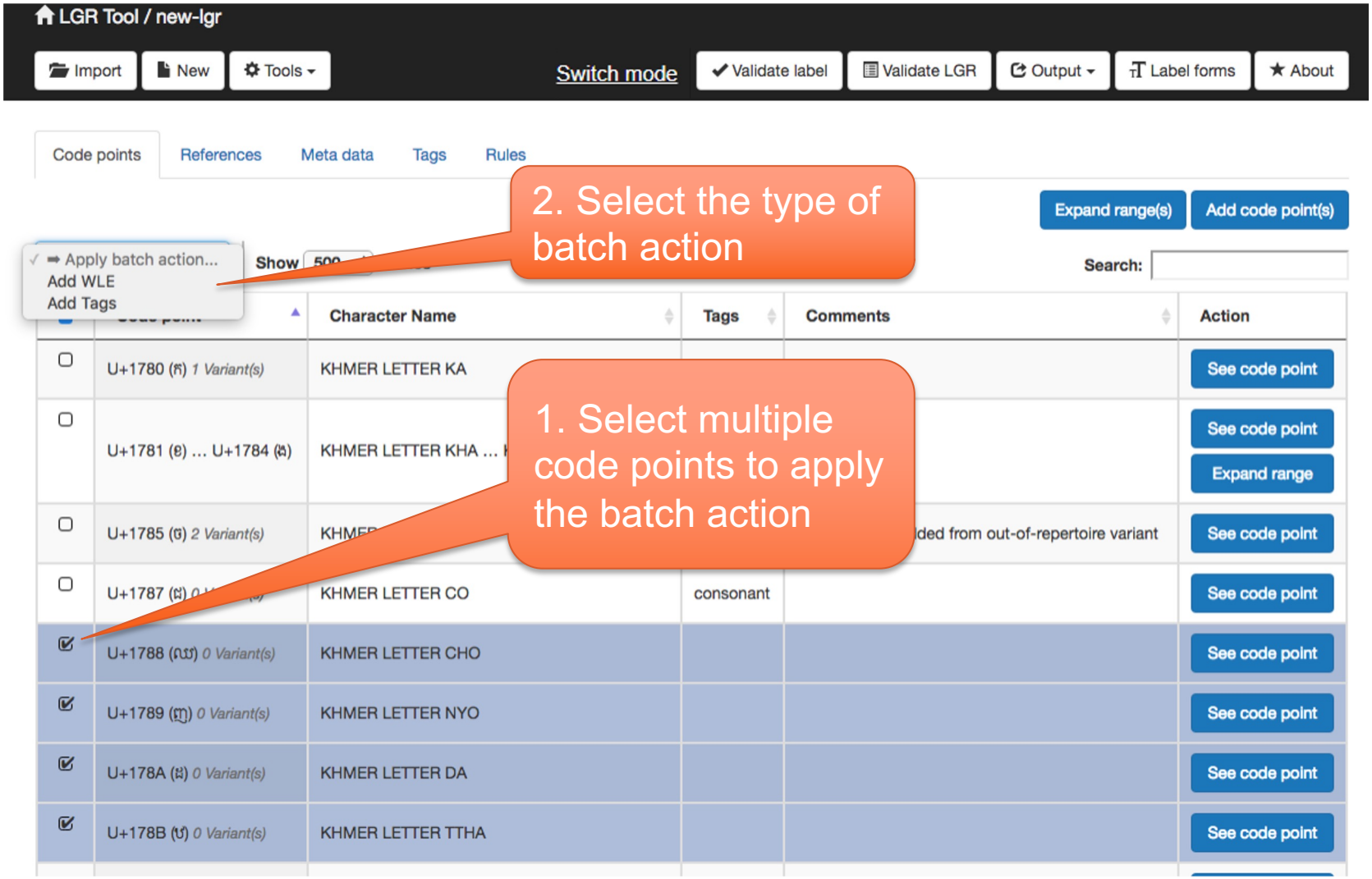
Define Rules



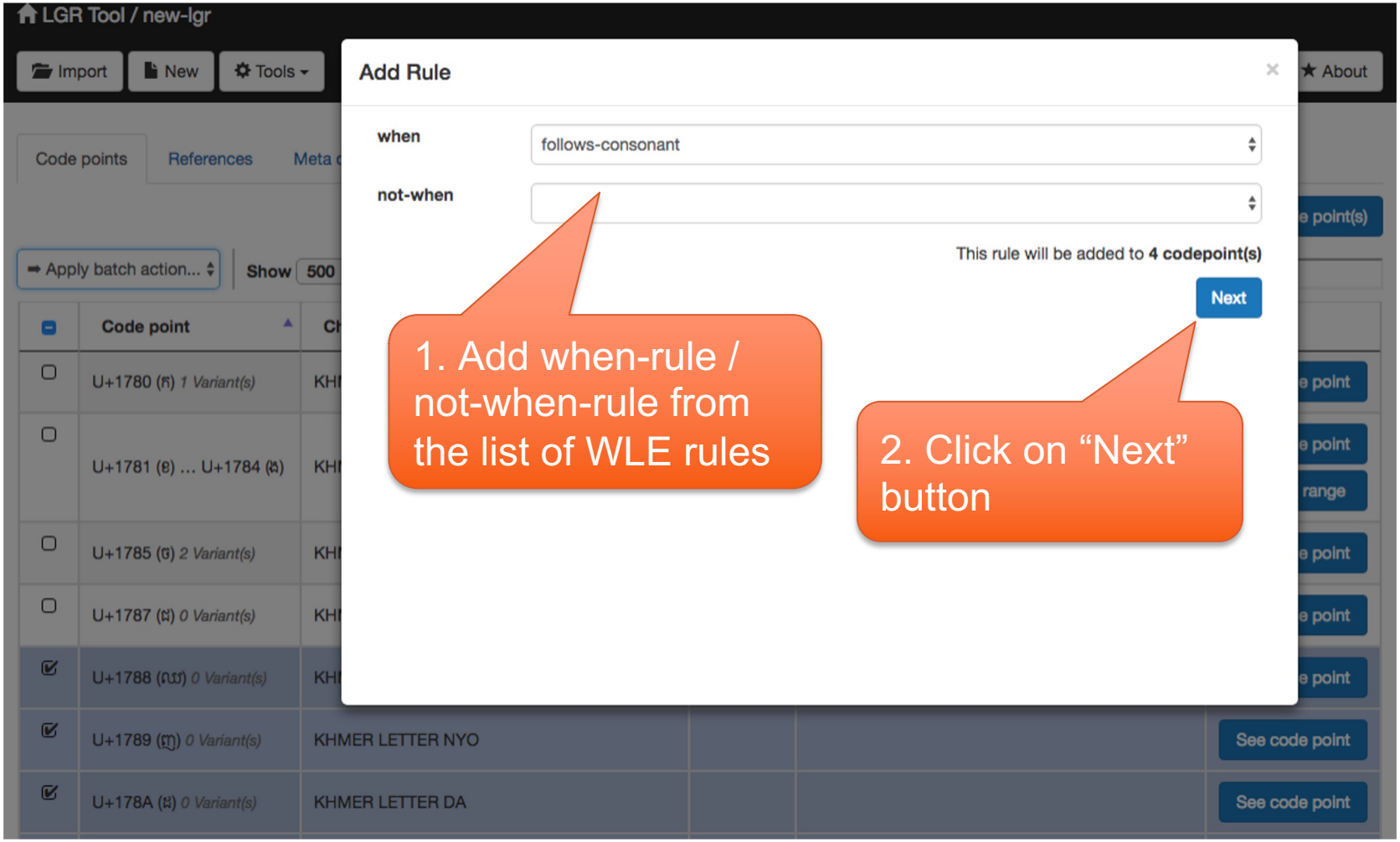
Define Actions



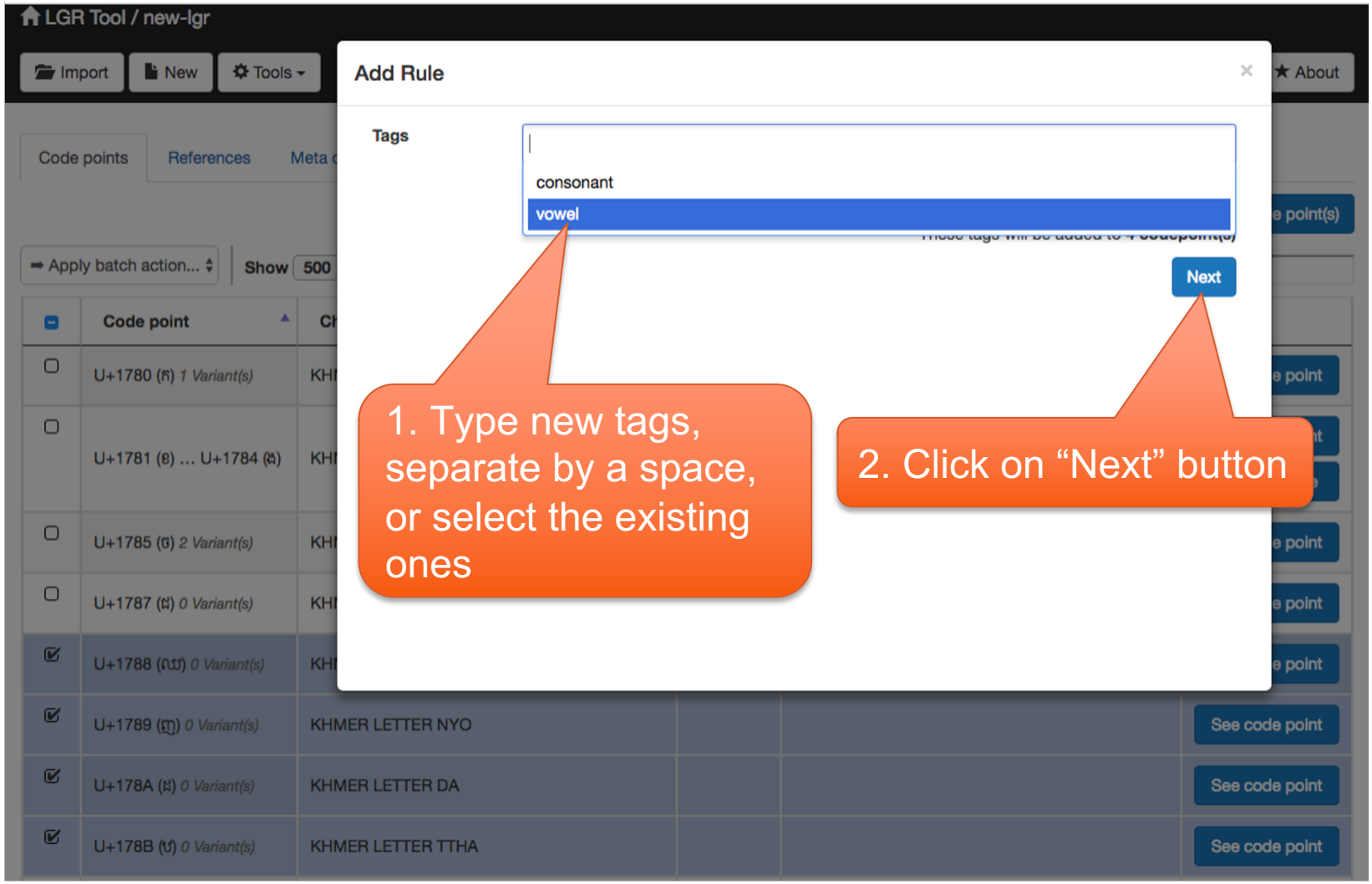
Apply Batch Action



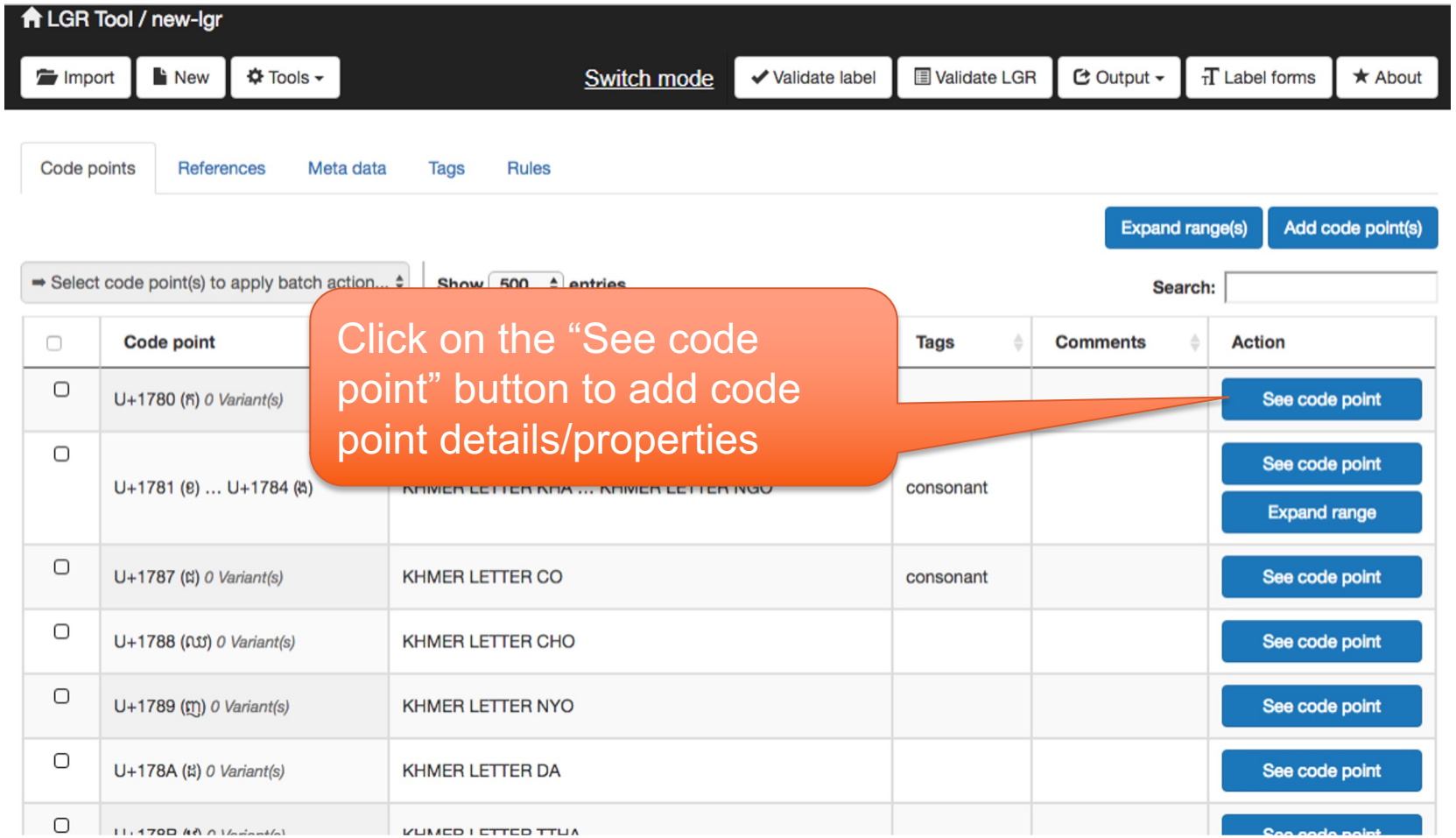
Apply Batch Action – Add WLE



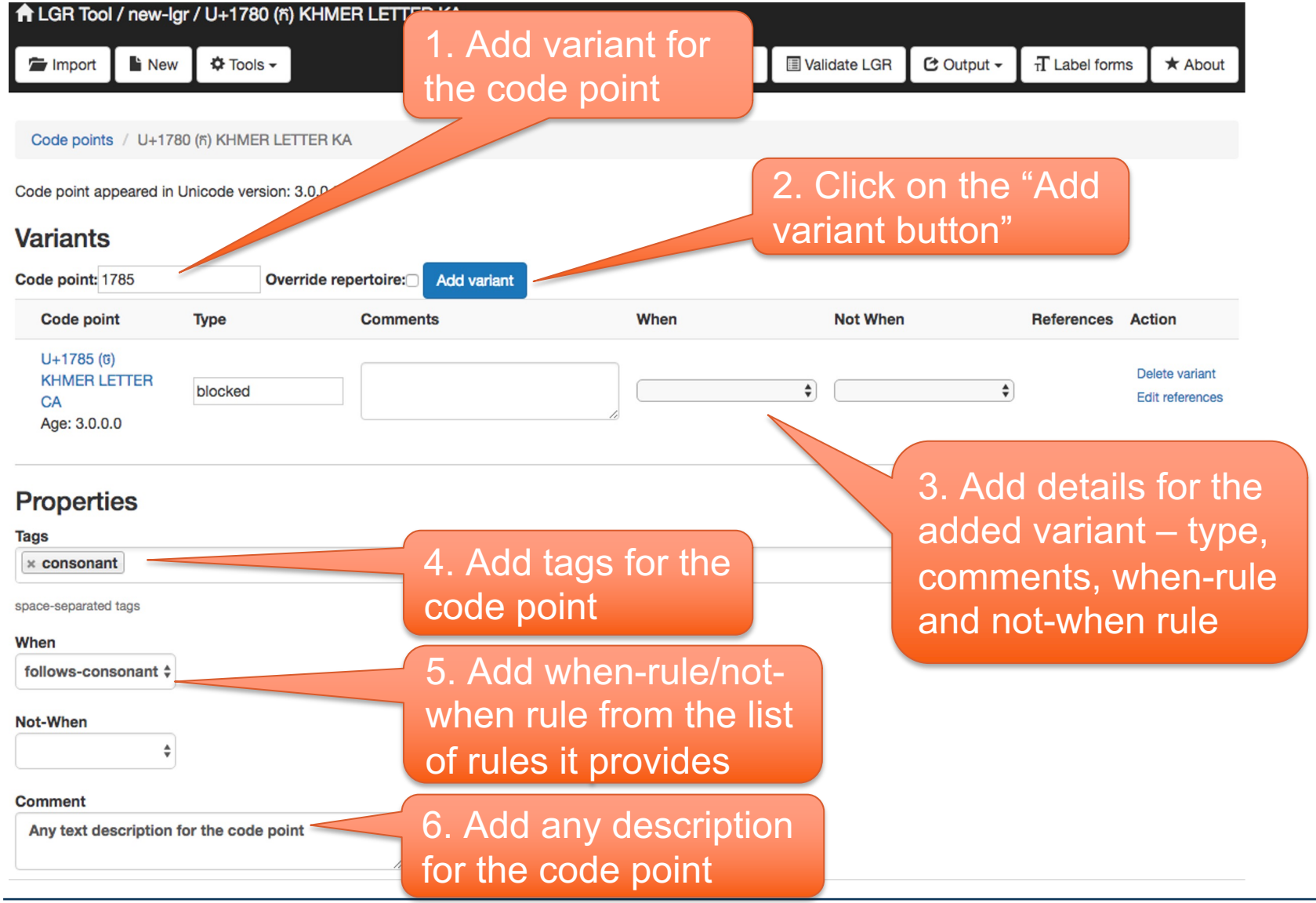
Apply Batch Action – Add Tags



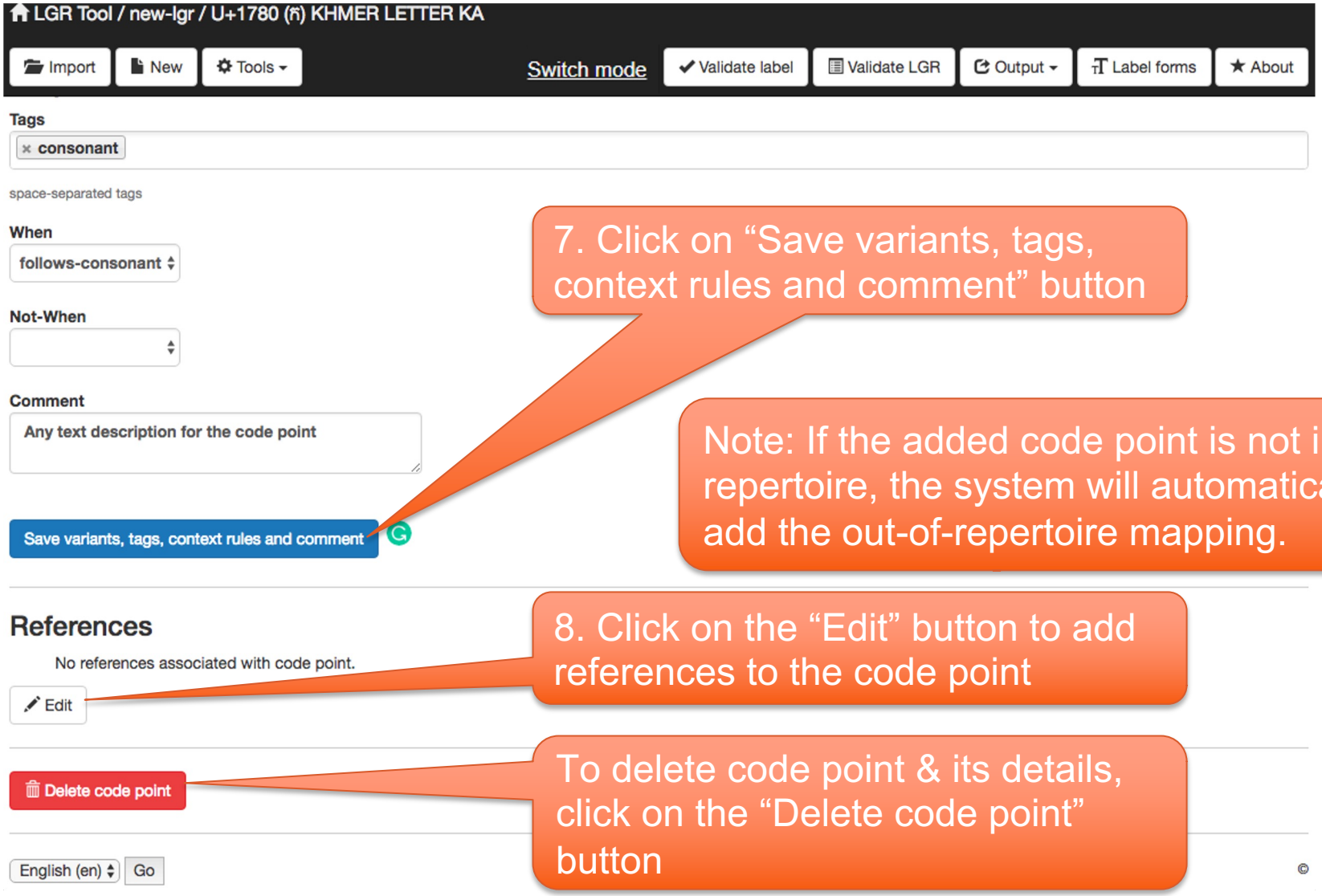
Define Code Point Properties



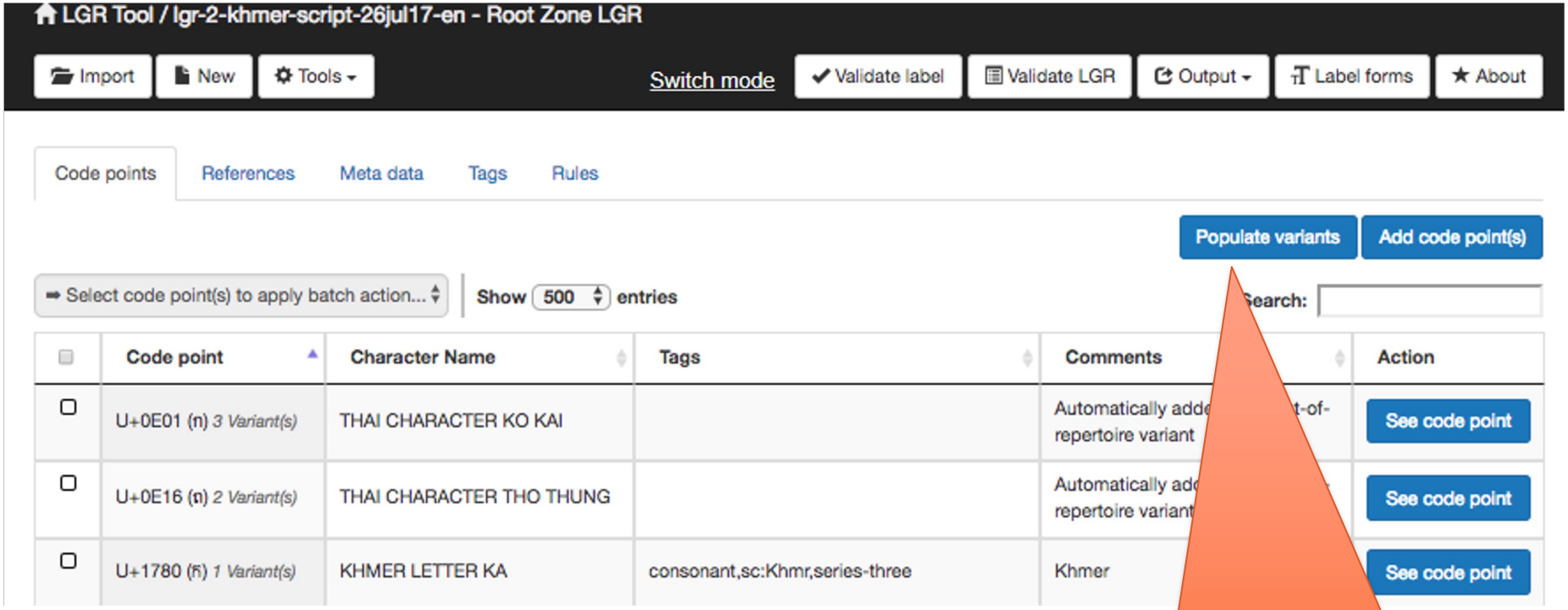
Define Code Point Properties - 1



Define Code Point Properties - 2



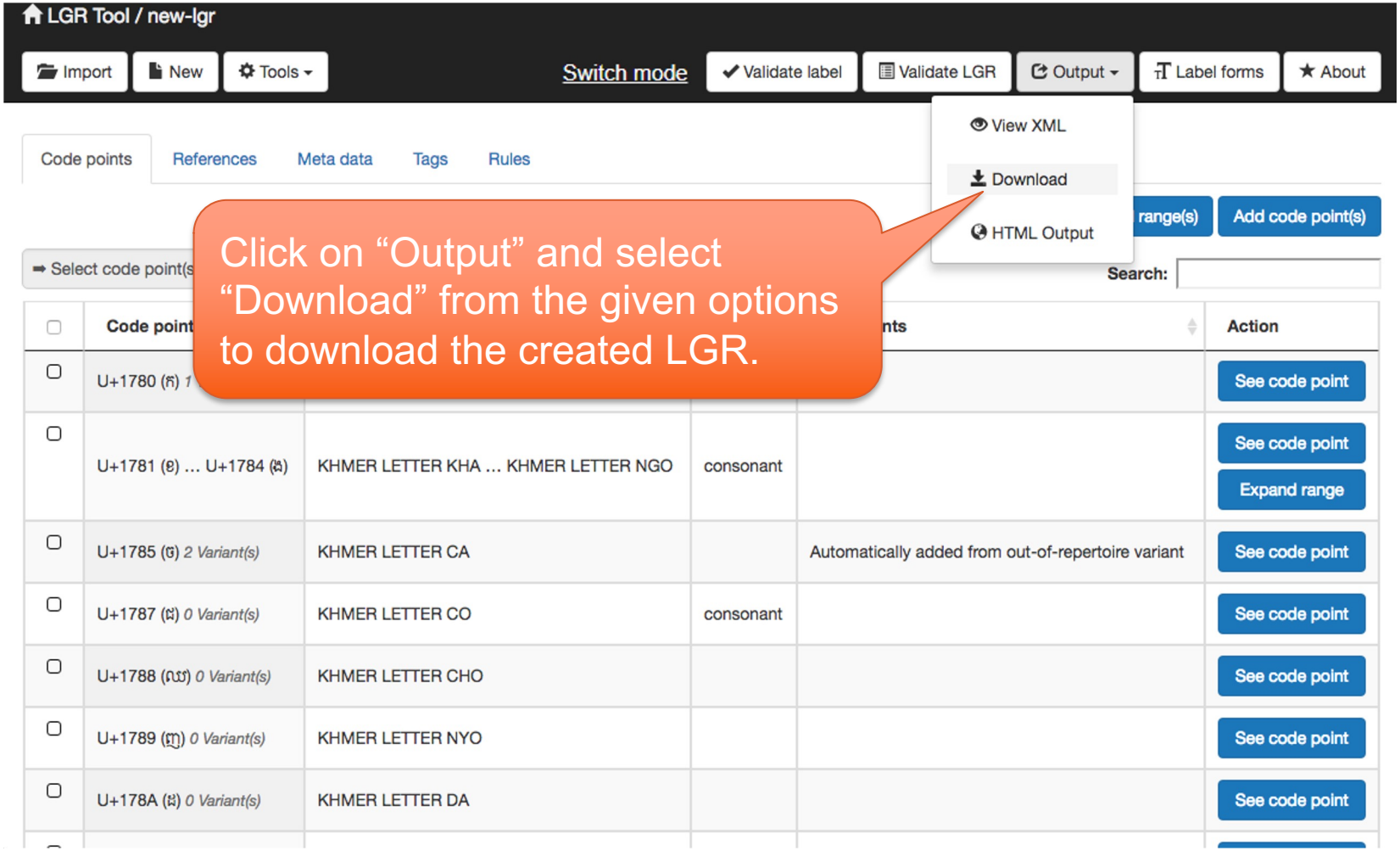
Define Code Point Properties - 3



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

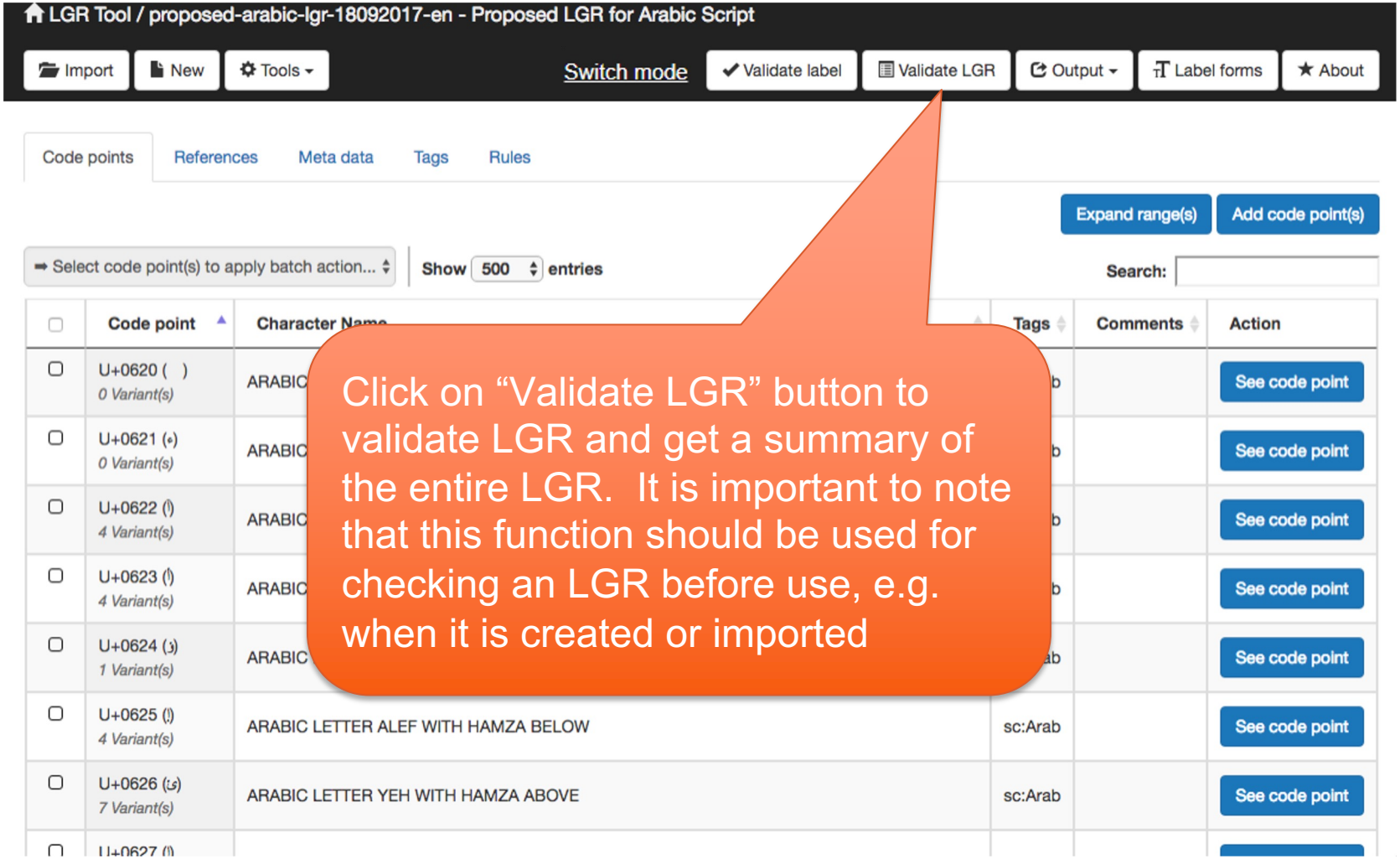
If the variant definitions are not symmetric, this button will show up. Click “Populate variants” to automatically populate variant mappings.

Download LGR

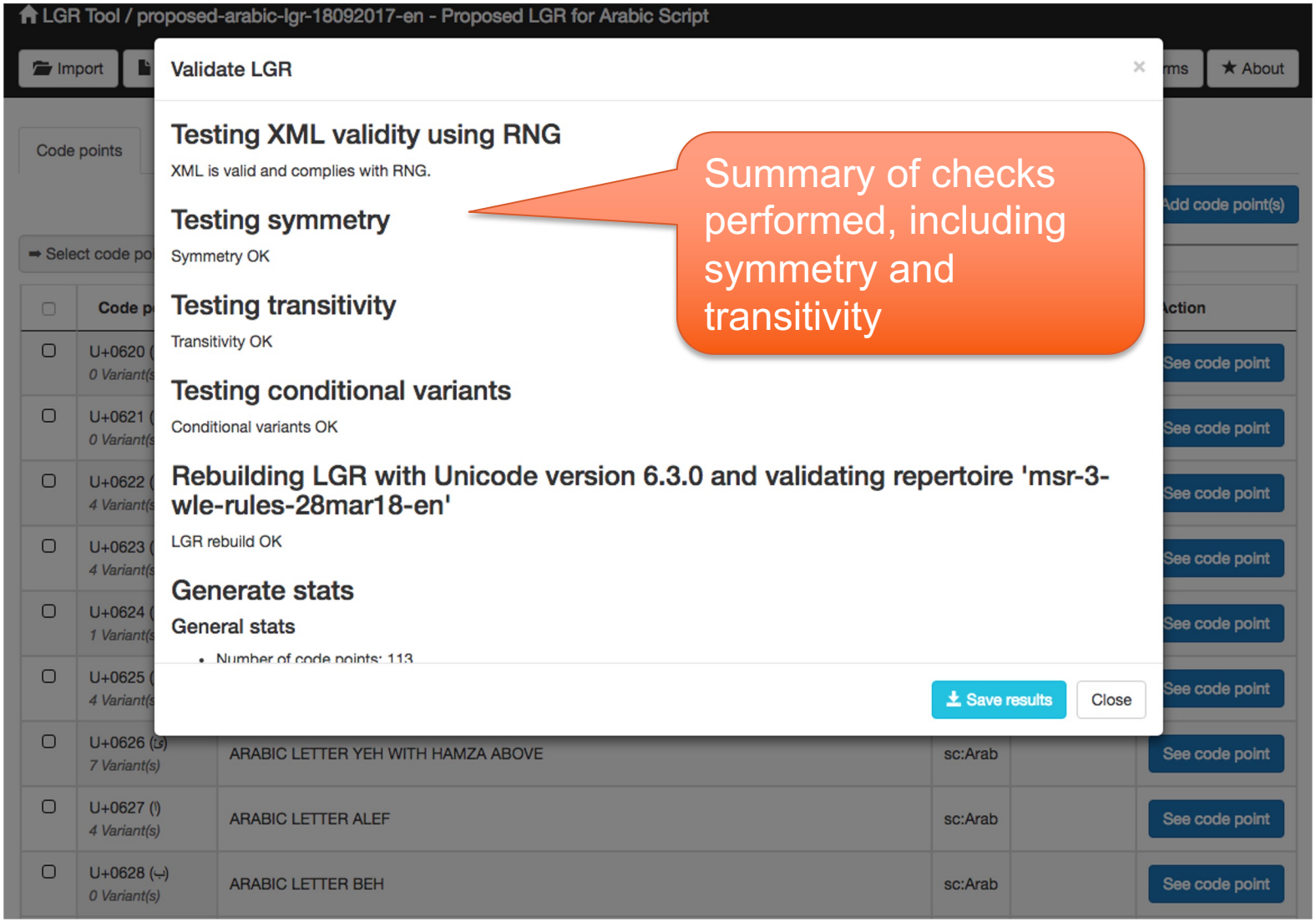


Validate LGR

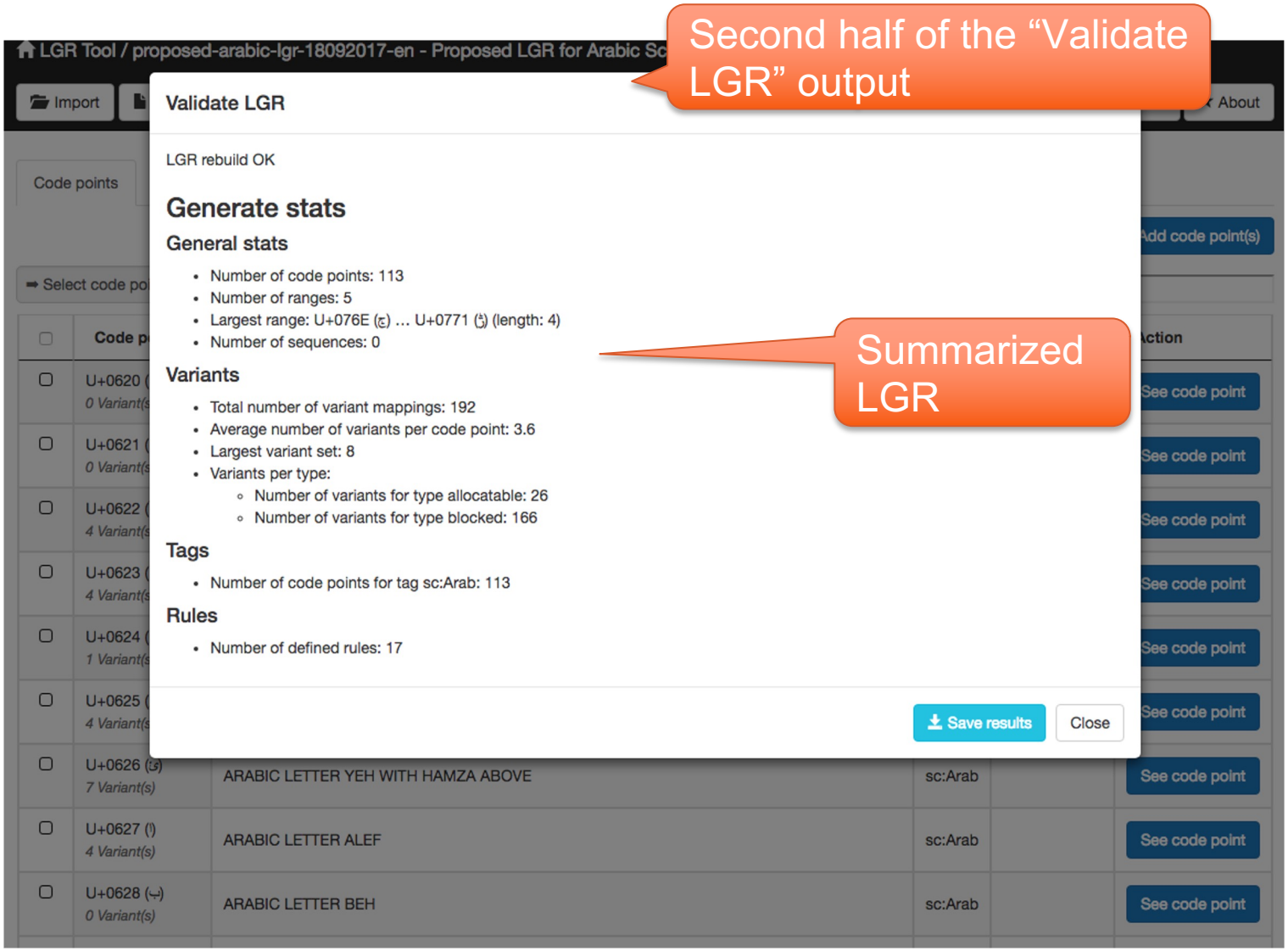
Validate LGR



Validate LGR

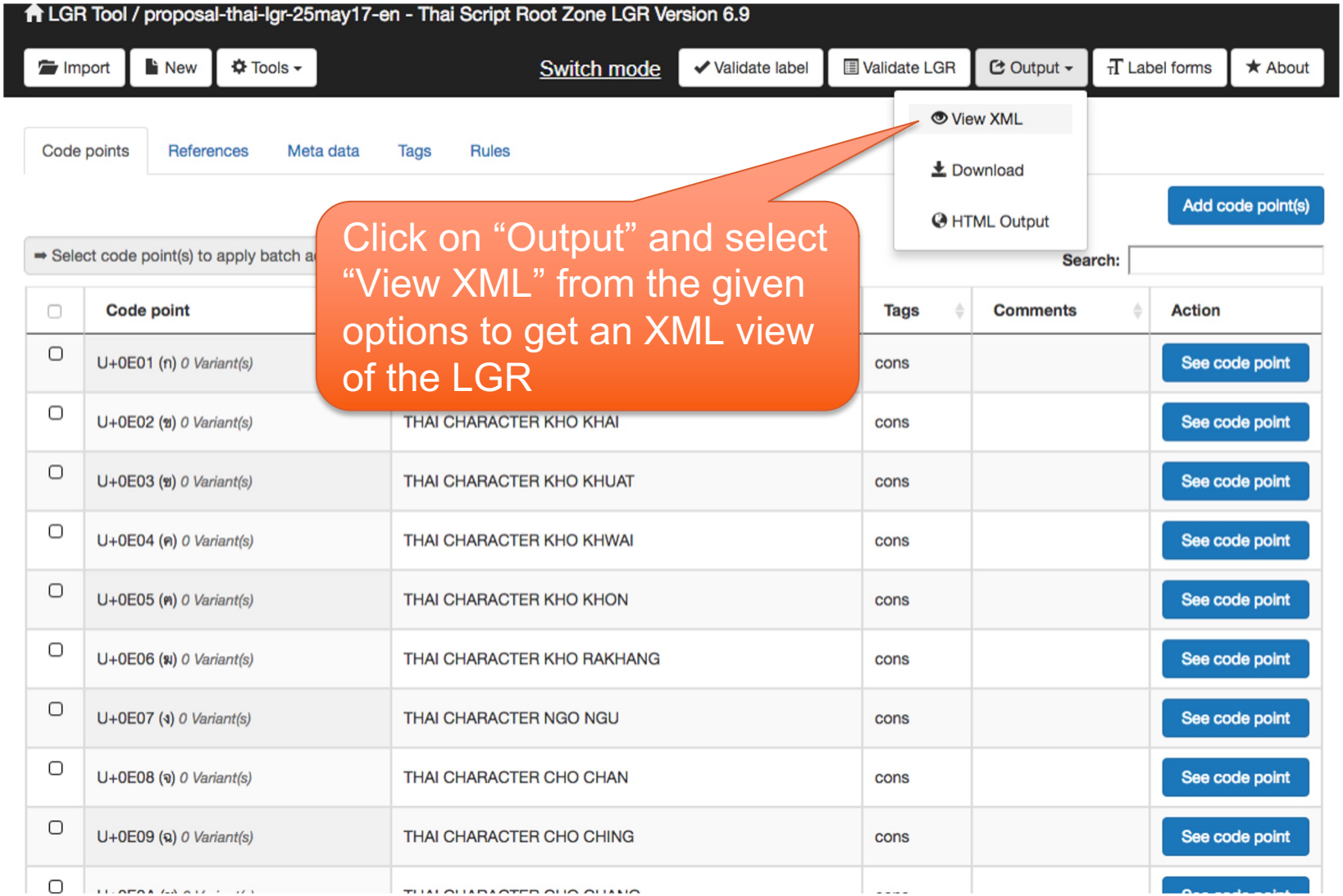


Validate LGR

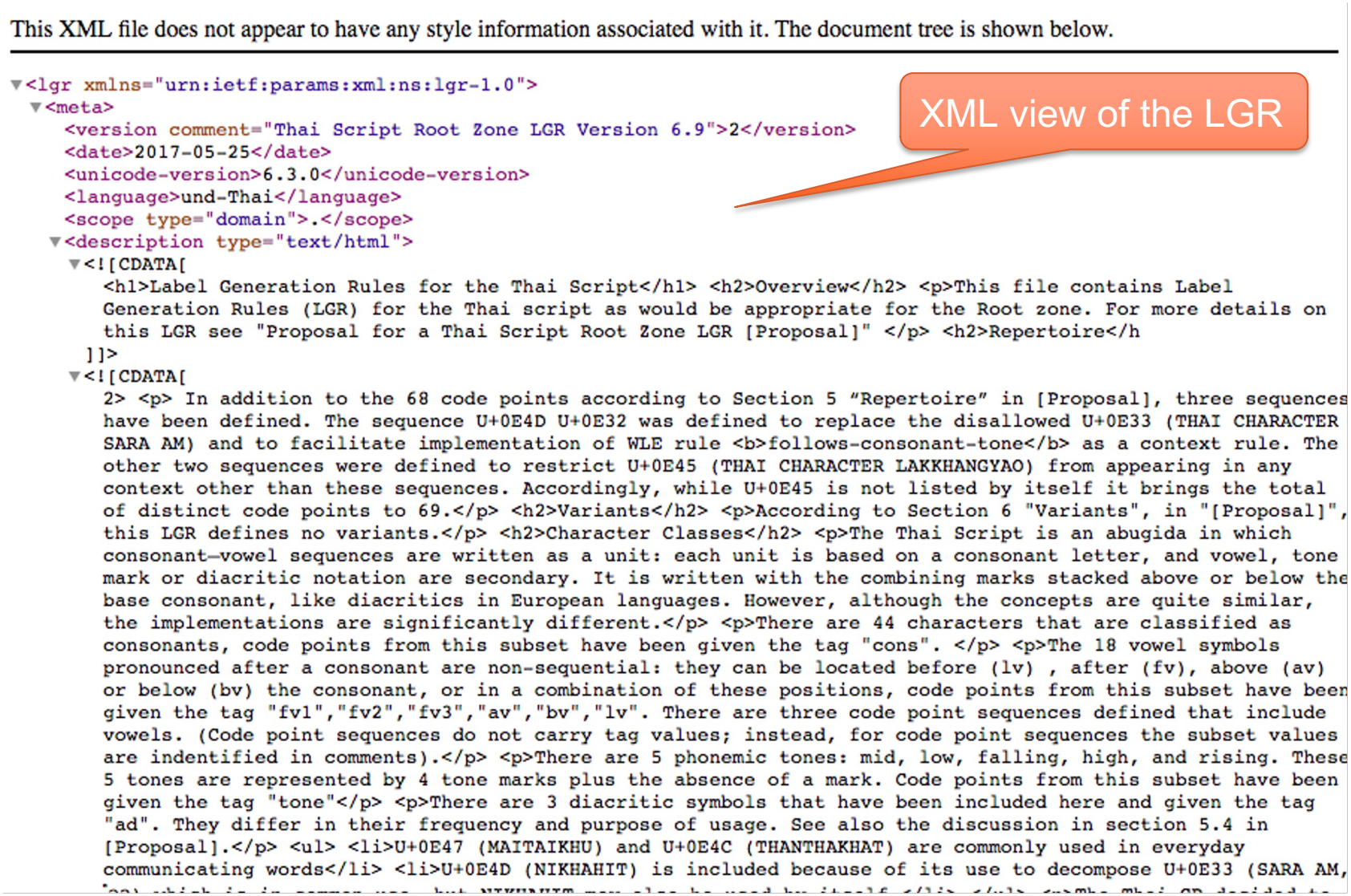


View LGR as XML and HTML

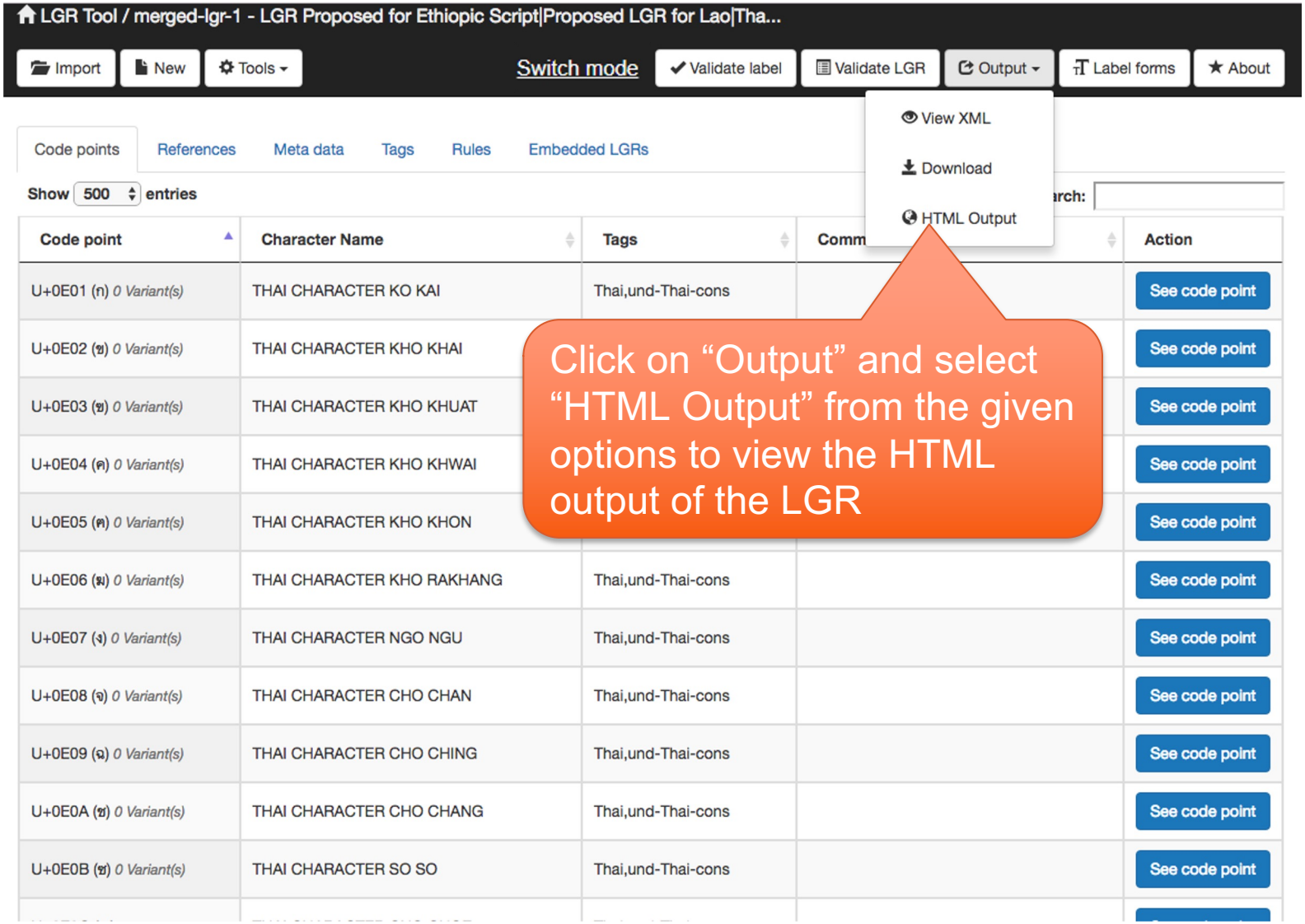
View LGR As XML



View LGR As XML



View LGR As HTML



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

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 indentified 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					

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	ሀ	U+1210	ሐ	↔	blocked	[4]	
2	U+1200	ሀ	U+1280	ኀ	↔	blocked	[4]	
3	U+1210	ሐ	U+1280	ኀ	↔	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-Laoo-Cf	Tag= **und-Laoo-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-Laoo-consonant	Tag= **und-Laoo-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-Laoo-semi-consonant	Tag= **und-Laoo-semi-consonant**	1	{U+0EBC}		
und-Laoo-tone-mark	Tag= **und-Laoo-tone-mark**	4	{U+0EC8 U+0EC9 U+0ECA U+0ECB}		
und-Laoo-vowel-above	Tag= **und-Laoo-vowel-above**	7	{U+0EB1 U+0EB4 U+0EB5 U+0EB6 U+0EB7 U+0EBB U+0ECD}		
und-Laoo-vowel-below	Tag= **und-Laoo-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:]∪[:und-Thai-above-vowel:]∪[: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:]∪[: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:]∪[:und-Thai-tone:]∪[: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:]u[: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:]I[: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:]I[:und-Laoo-semi-consonant:]I[:und-Laoo-tone-mark:]I[: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:]I[:und-Laoo-semi-consonant:]I[:und-Laoo-vowel-above:]I[: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://lexitron.nectec.or.th/KM_HL5001/file_HL5001/Paper/Inter%20Journal/krrp_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)

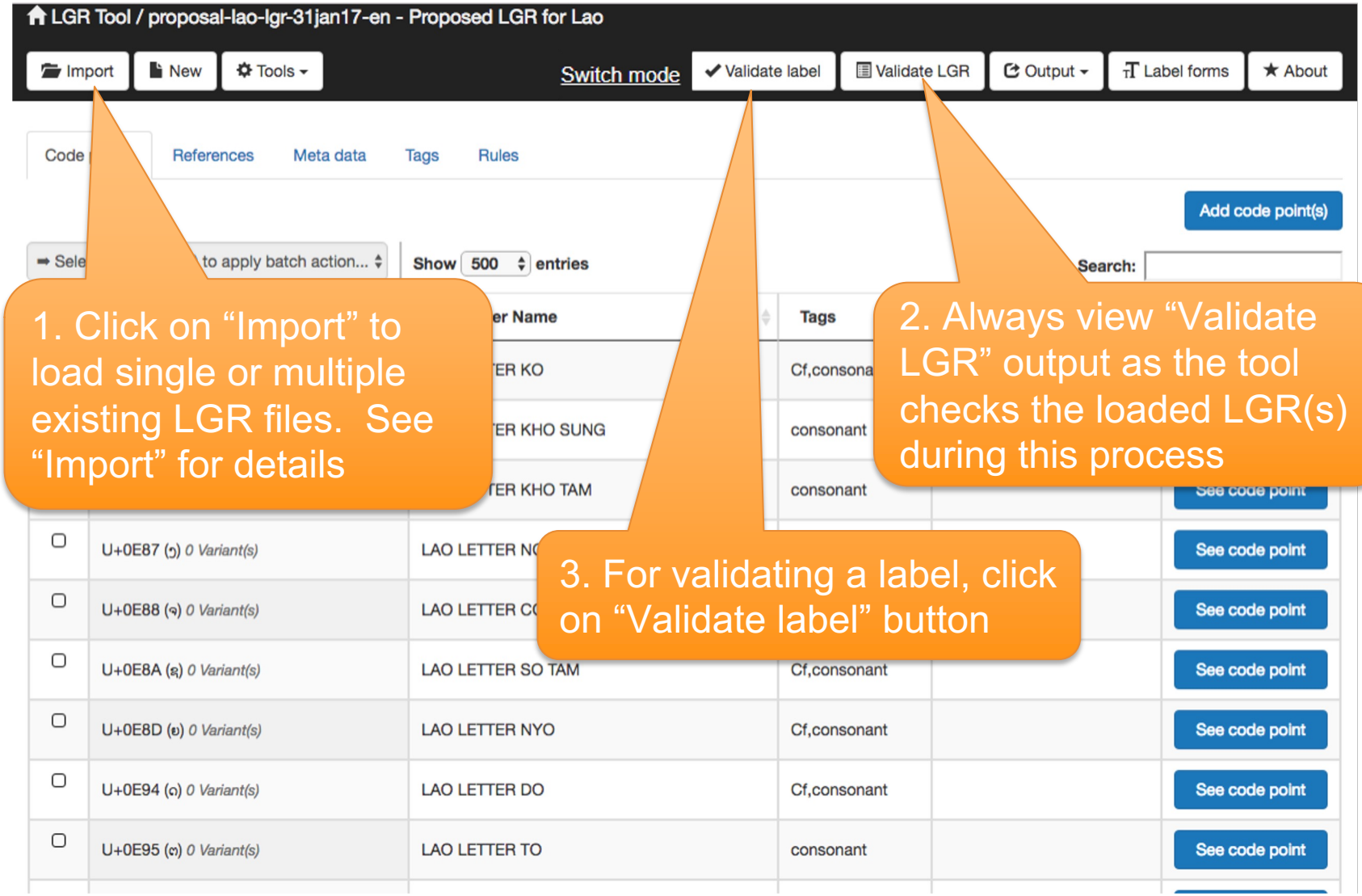
[201] Lao grammar book published by the Ministry of Education in 1967, see Appendix B, Figure 1

[202] Lao grammar book published by the Ministry of Education in 1967, see Appendix B, Figure 2

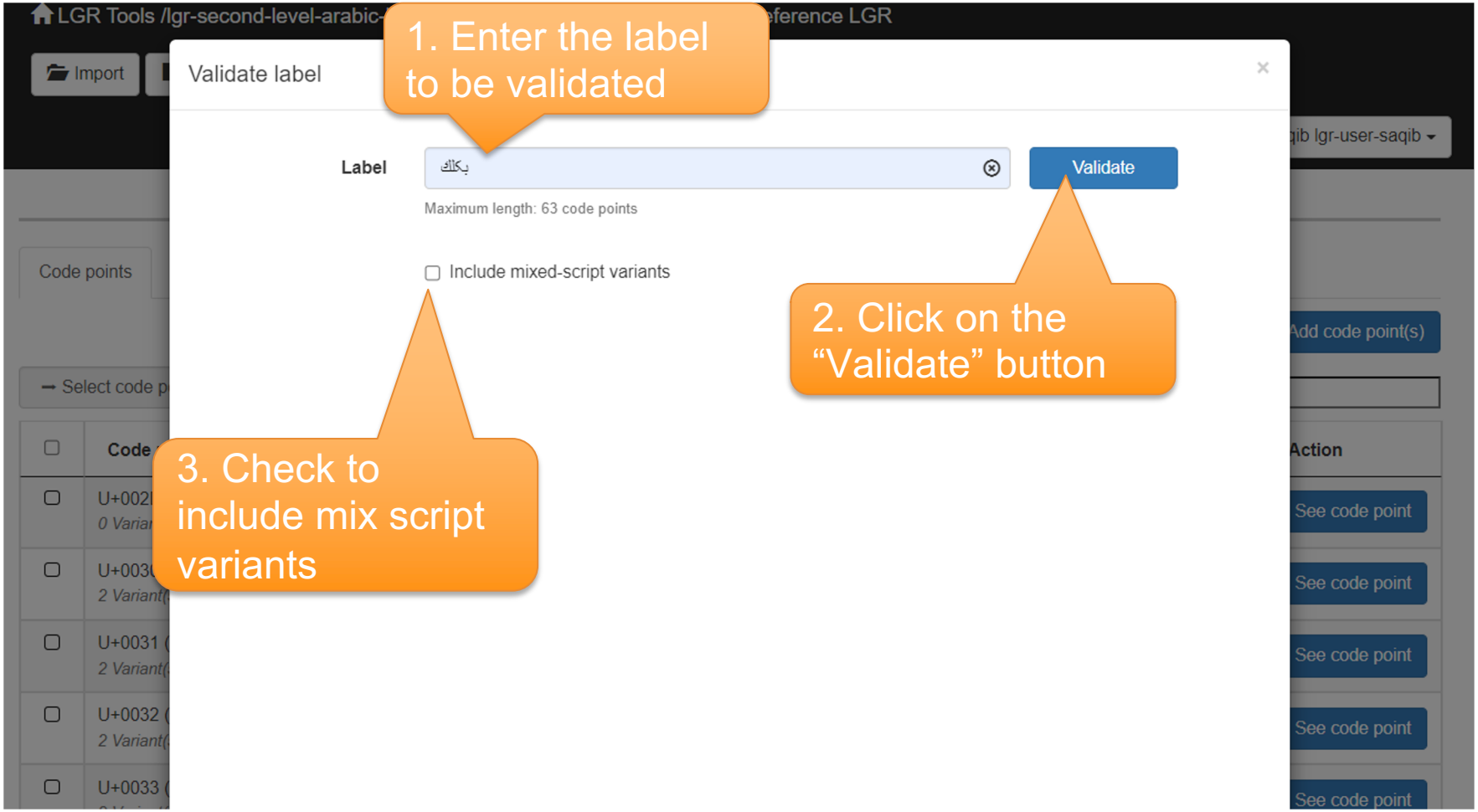
[203] Lao grammar book published by the Ministry of Education in 1967, see Appendix B, Figure 3

Validate a Label

Validate a Label

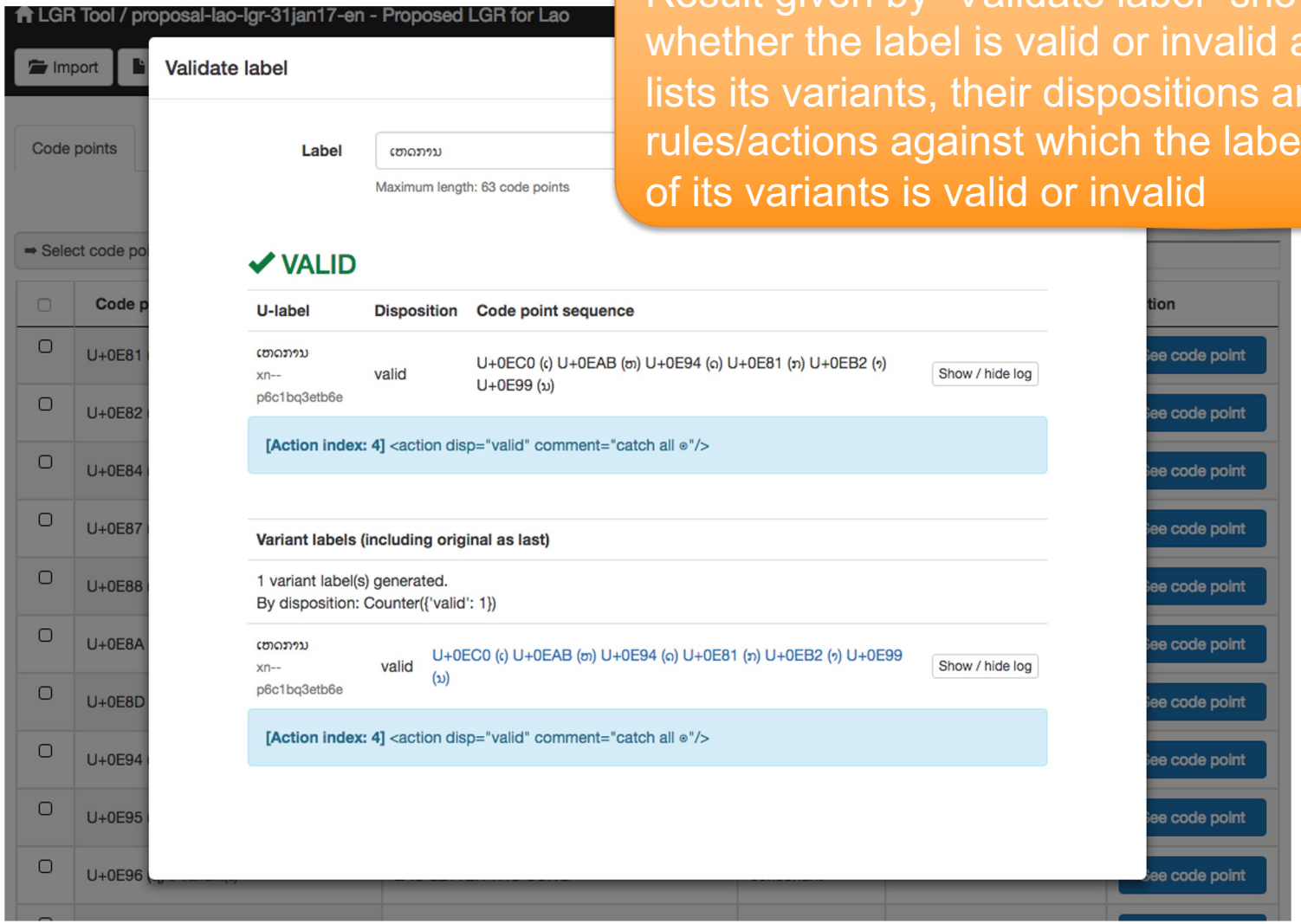


Validate a Label with a Single LGR

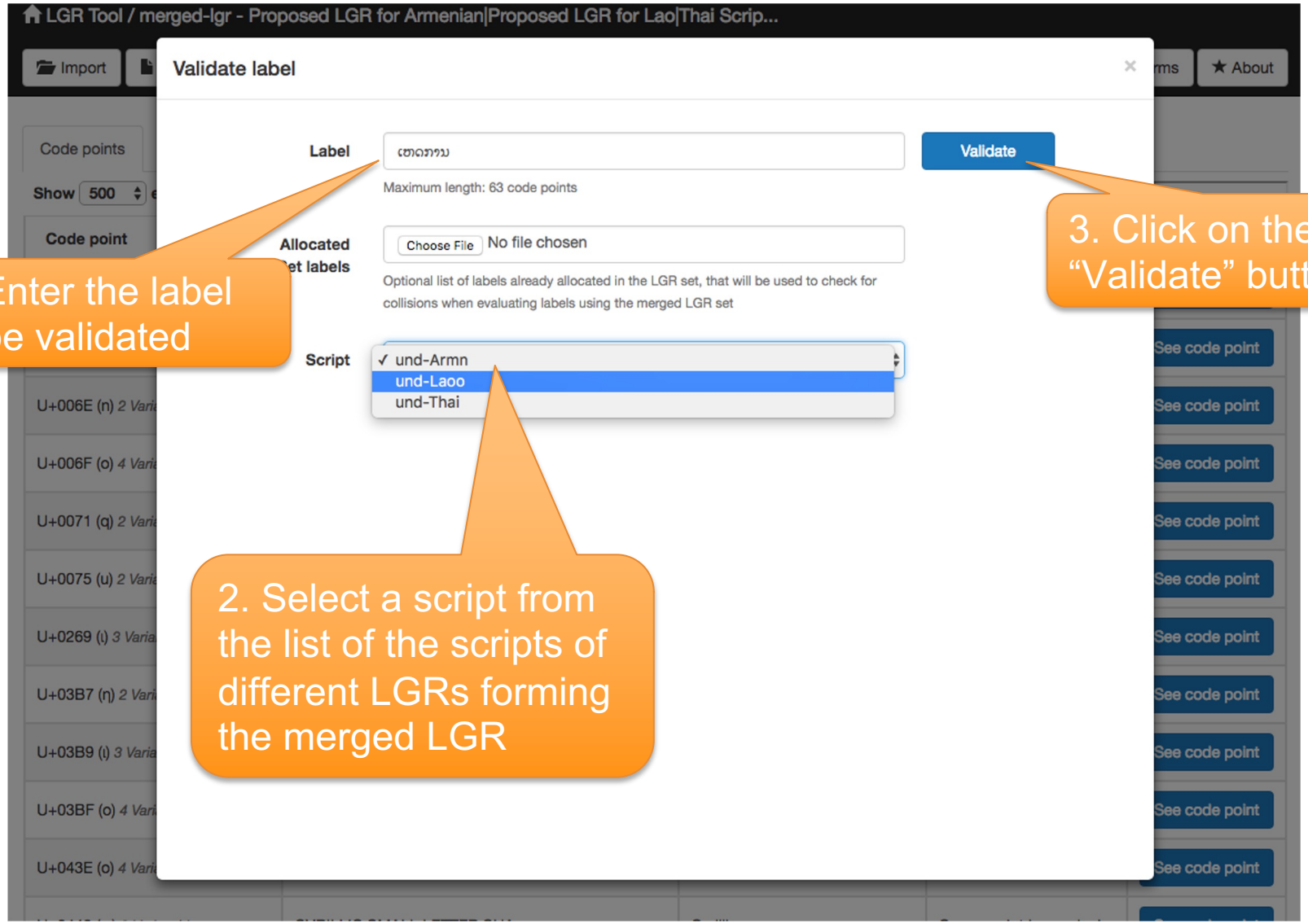


Validate a Label with a Single LGR

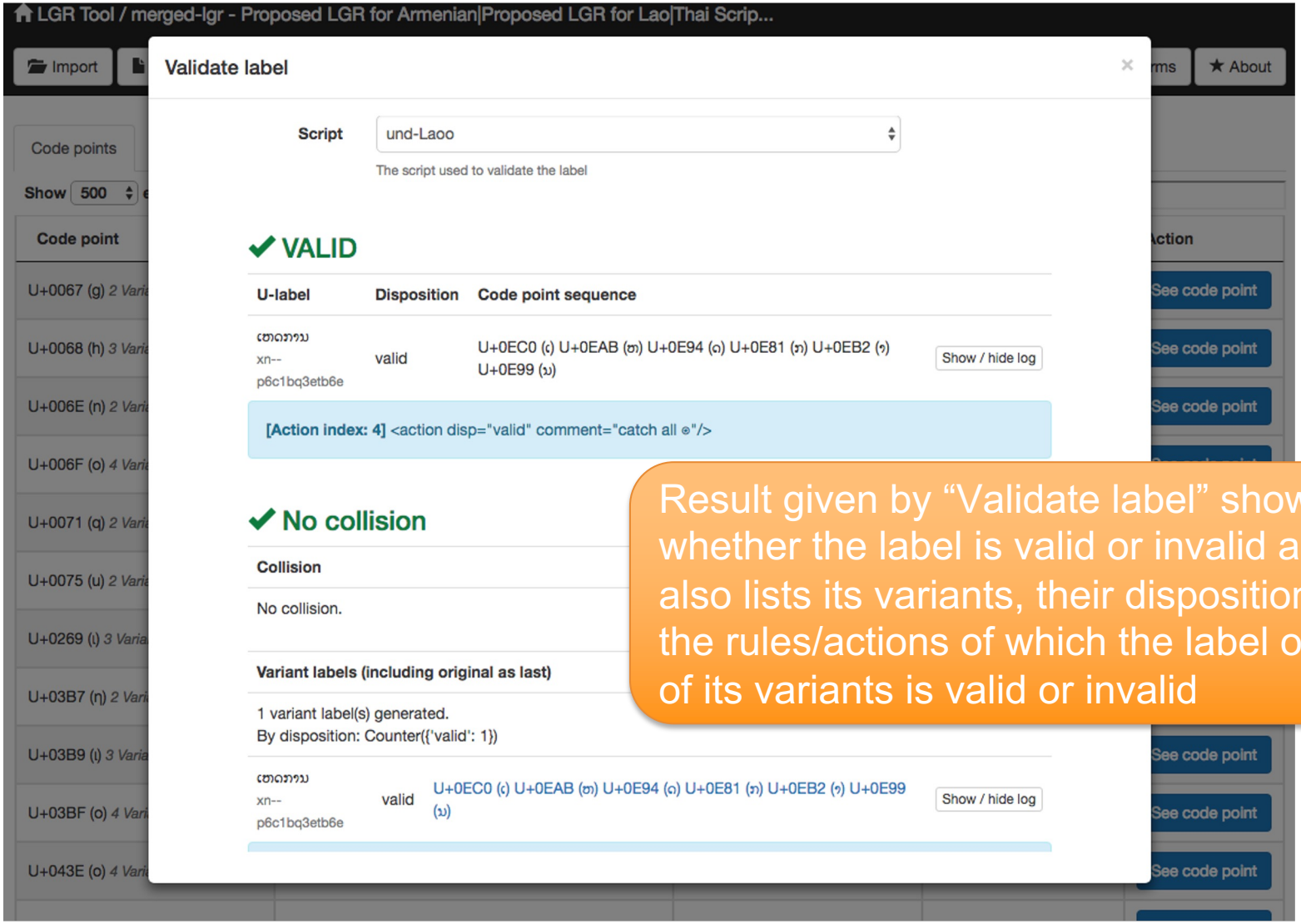
Result given by “Validate label” shows whether the label is valid or invalid and also lists its variants, their dispositions and the rules/actions against which the label or any of its variants is valid or invalid



Validate a Label with a Merged LGR



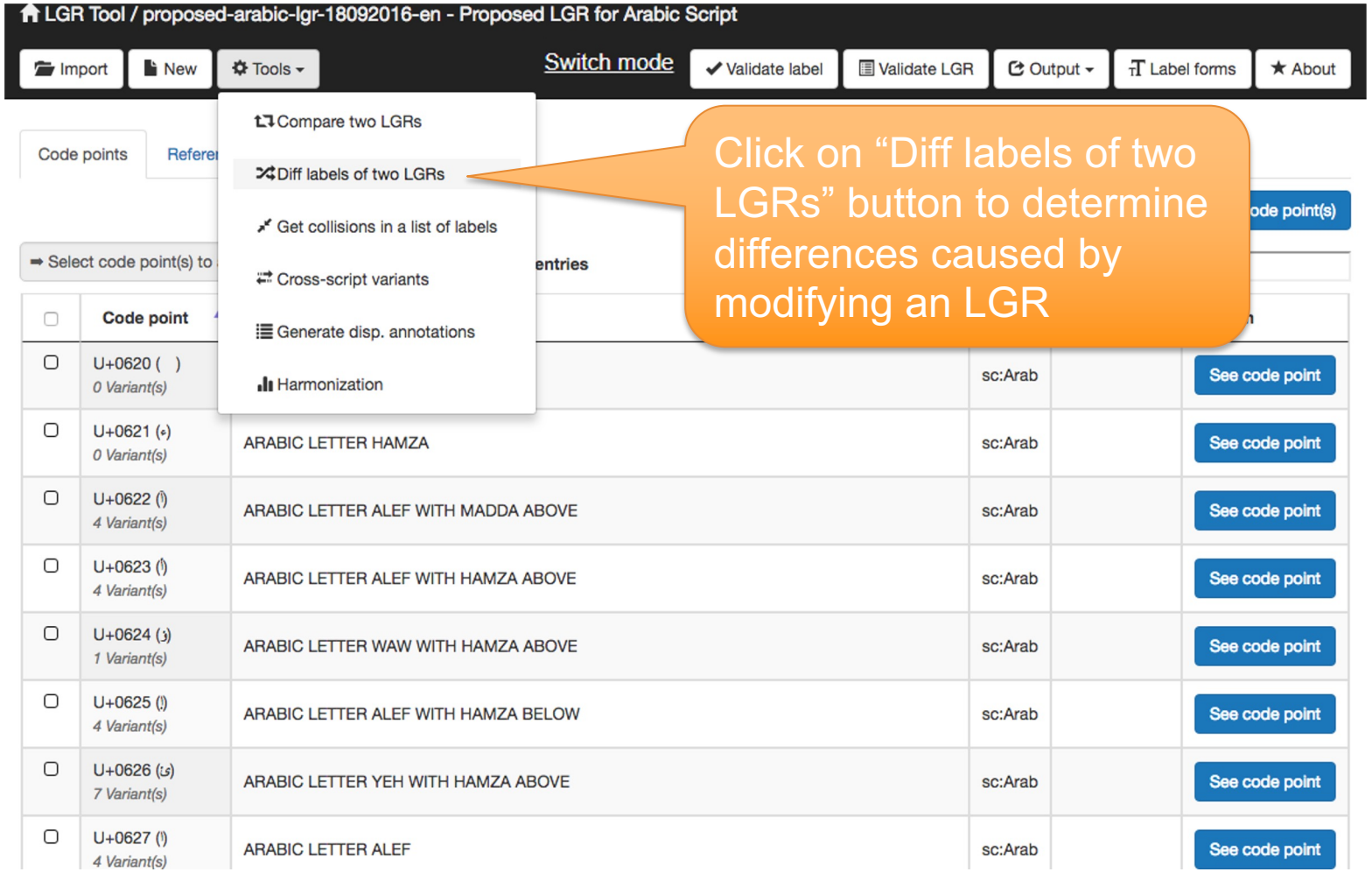
Validate a Label with a Merged LGR



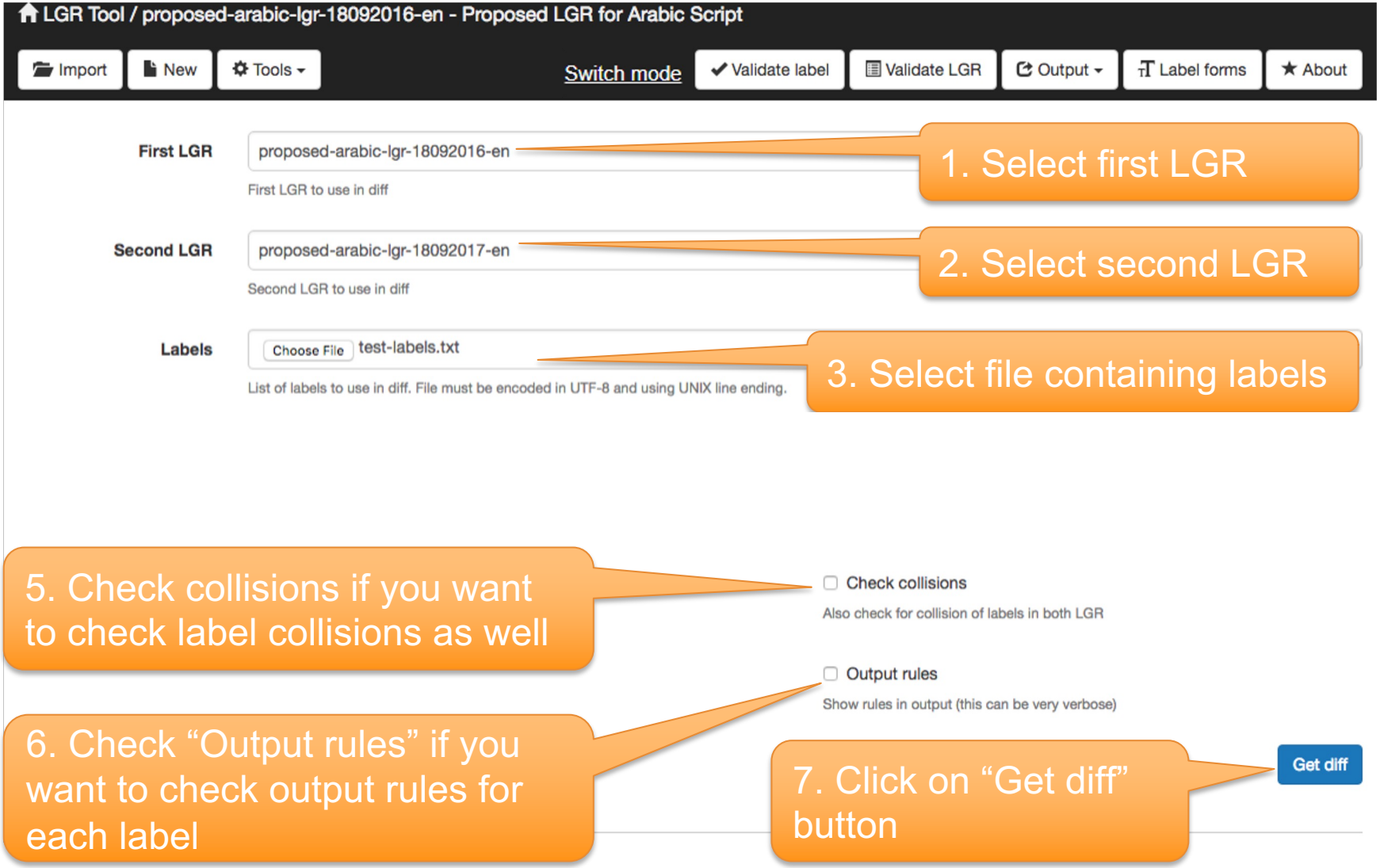
Result given by “Validate label” shows whether the label is valid or invalid and also lists its variants, their dispositions and the rules/actions of which the label or any of its variants is valid or invalid

Review Impact on Existing Labels by Revising an Existing LGR

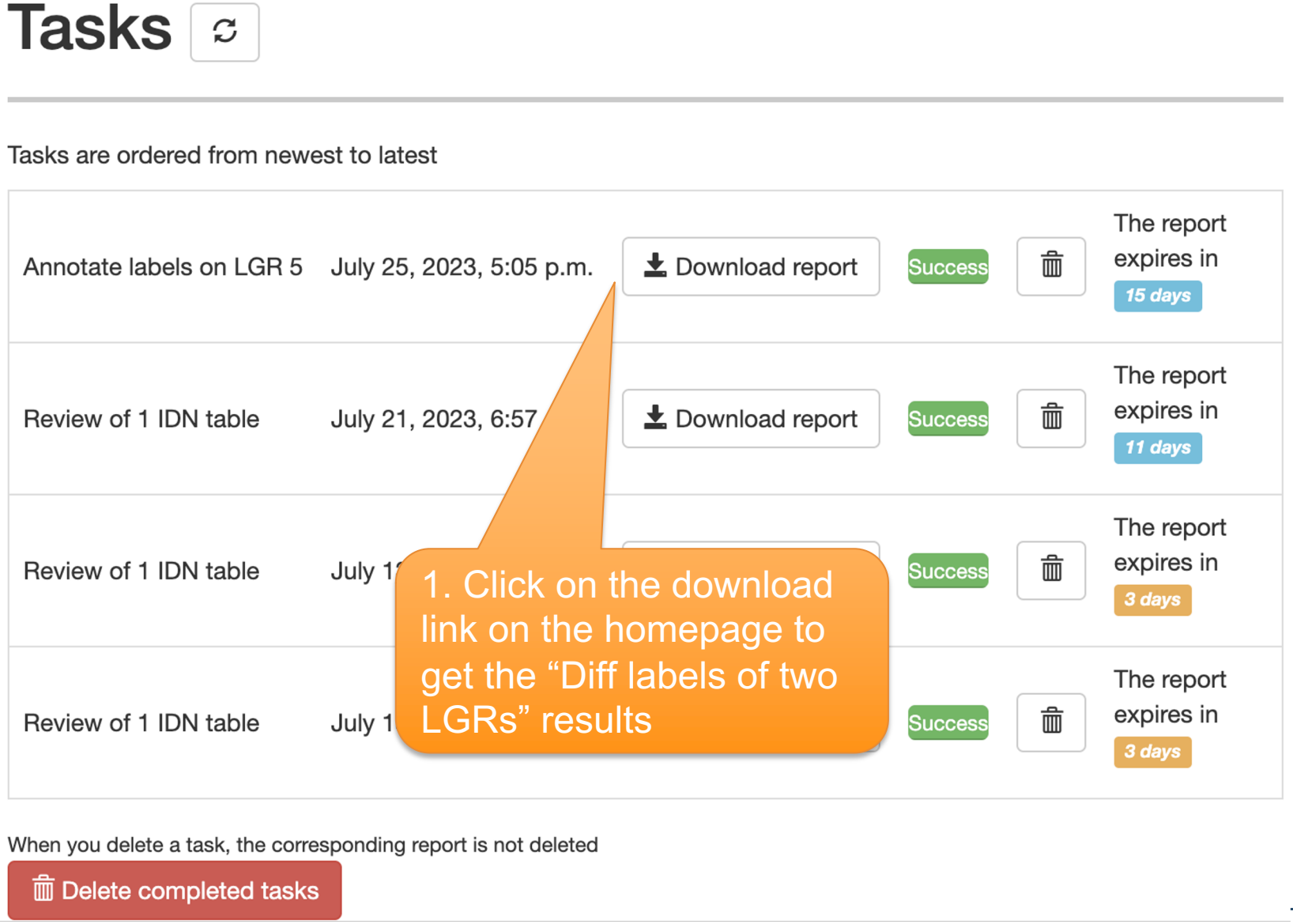
Review Impacts on Existing Labels Caused by Reviewed LGR



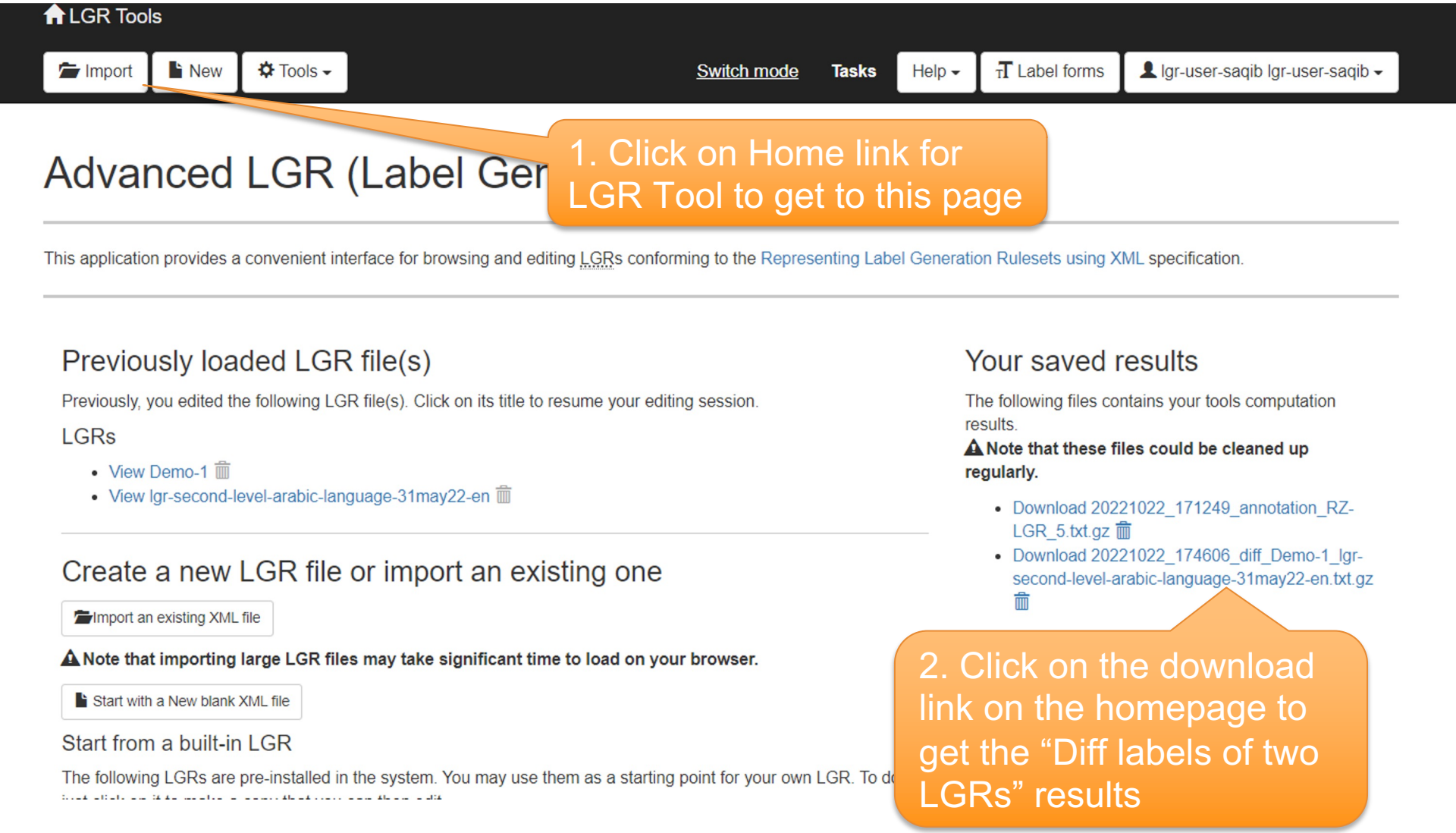
Enter Details



Download Results from Task Status Page



Download Results



Results

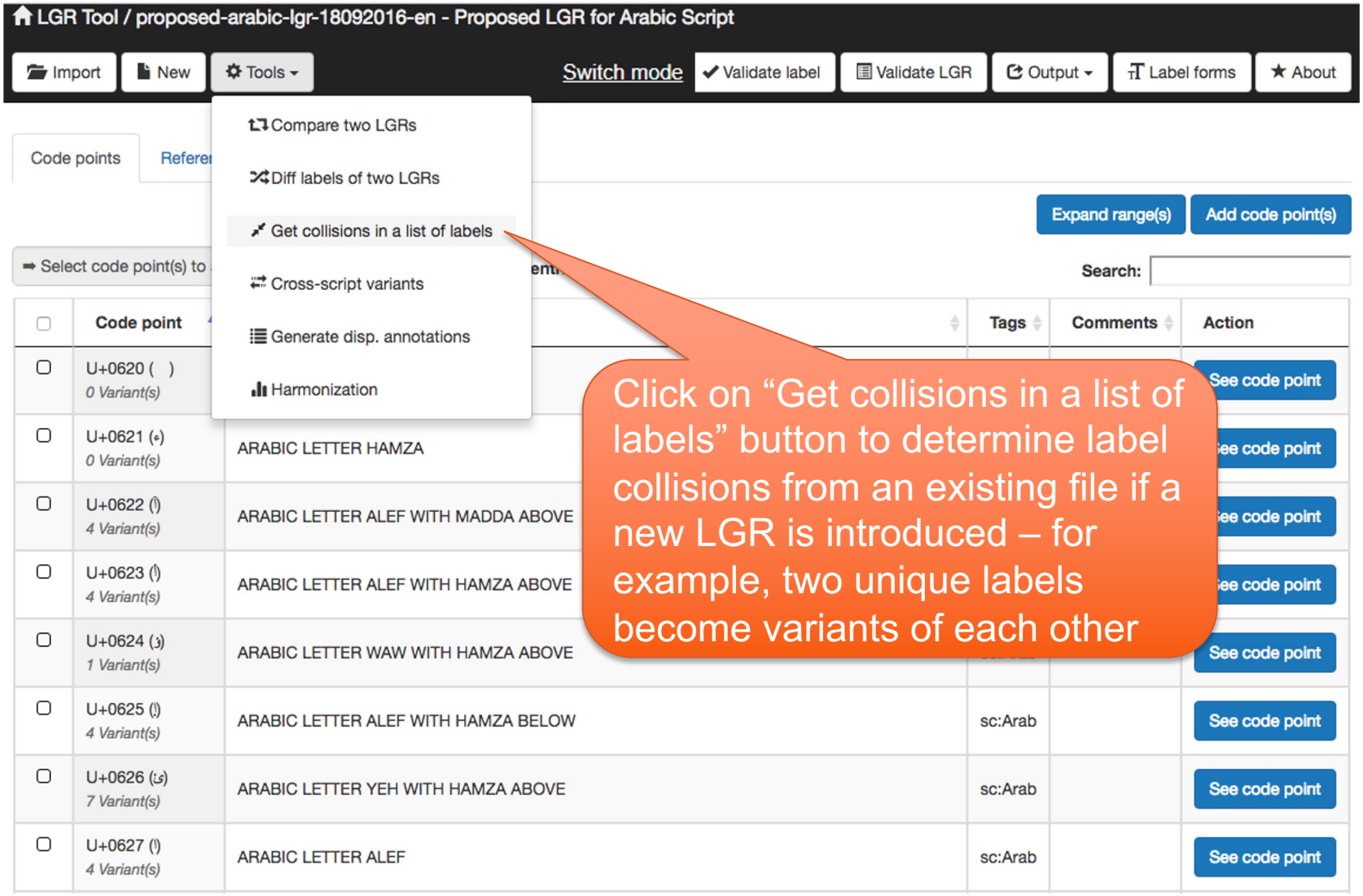
```
# Labels not in LGR 1 #


# 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 'كلكتم' [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

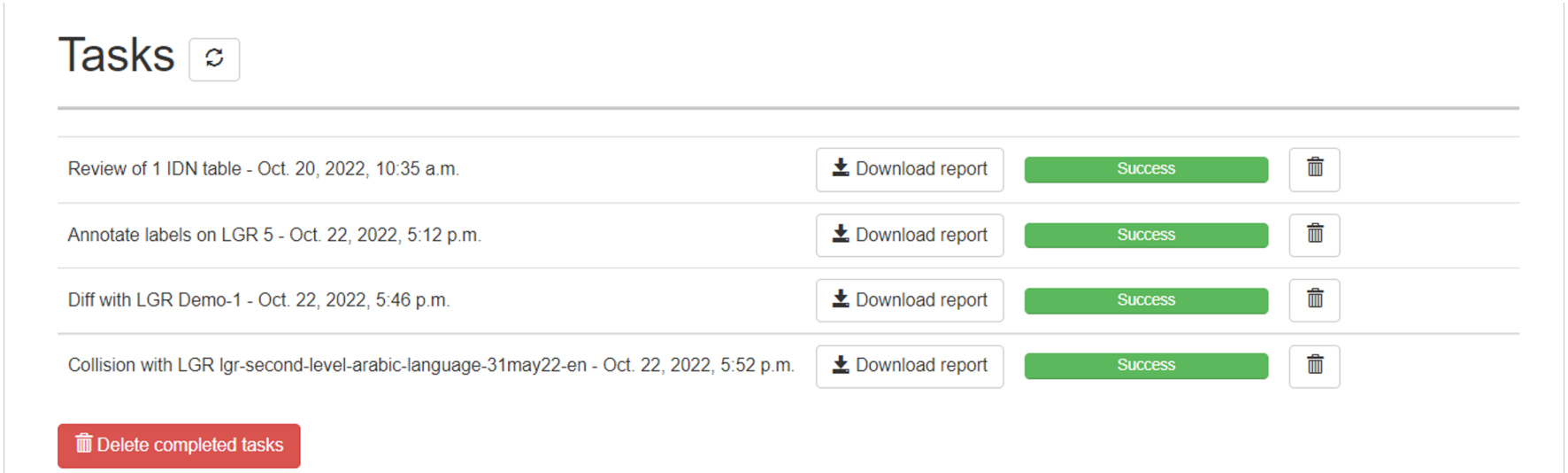
Review Impact on Existing Labels by Introducing a New LGR

Select Get Collisions in a List of Labels



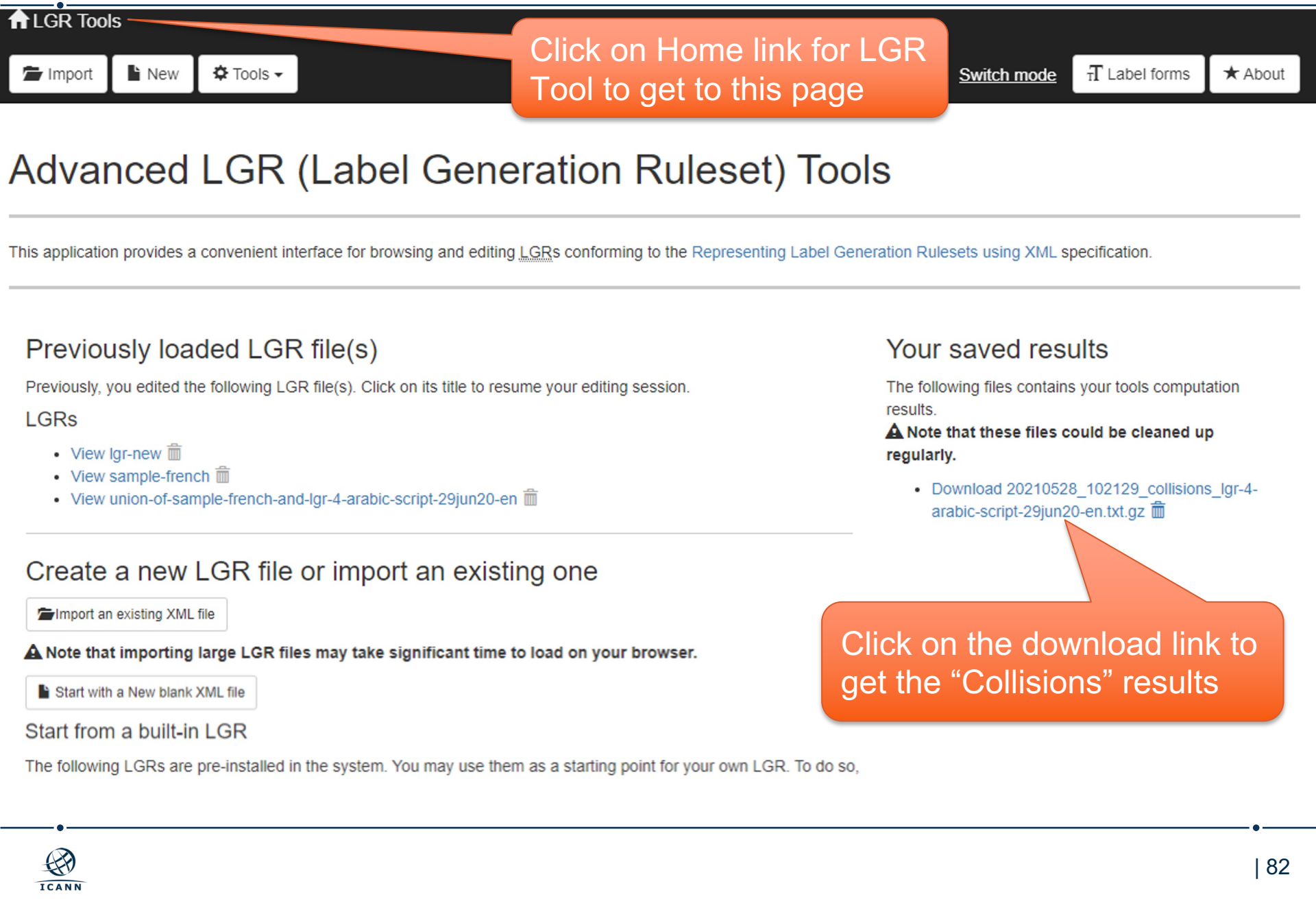
Enter Details

Download Results from Task Status Page



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

Download Results



Results

```
# Labels not in LGR #

Label مجموعة
Label يهدد
Label مرفق
Label مفعول
Label معضلة
Label لجنة
Label محمد
Label موعد

# Collisions #


## Collision ##

```
Label:          'كتكتم' |                                  'عتكتم'
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:          'عتكتة' |                                  'عتكتم'
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 'عتكتم' [U+06AA U+0644 U+0643 U+062A U+06C1]

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

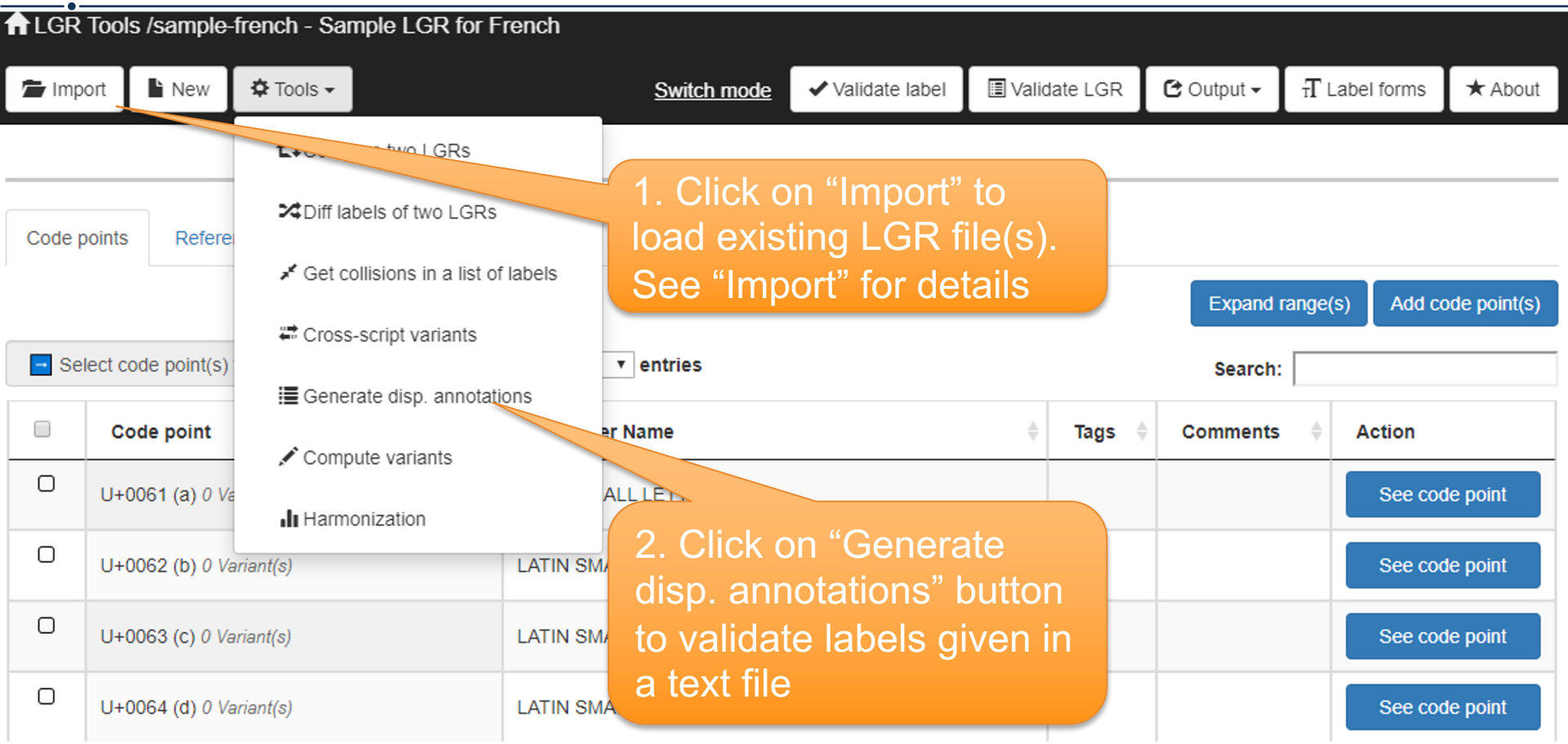
```

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

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

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

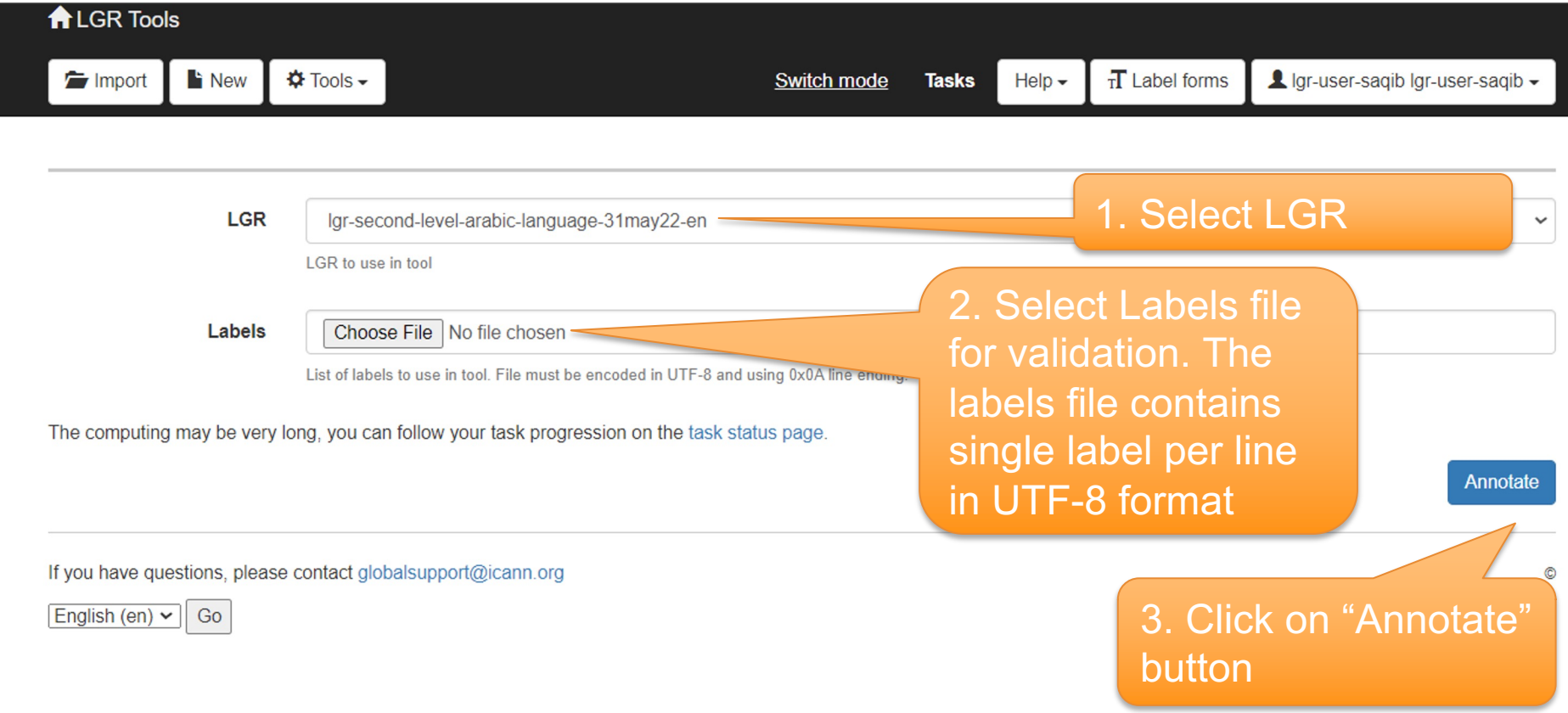
Details of a label and its corresponding variant(s)

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

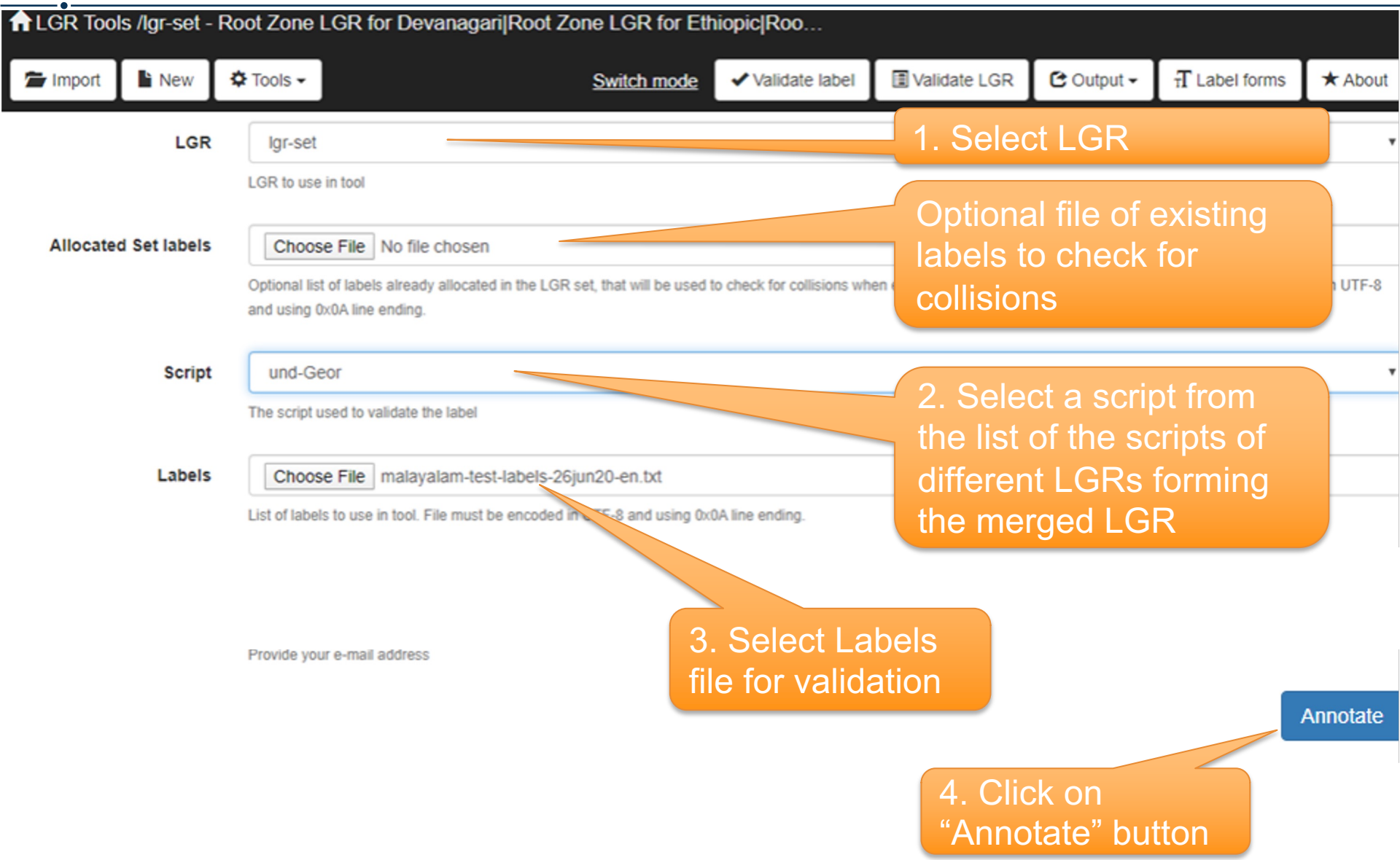
Select Generate Disposition Annotations



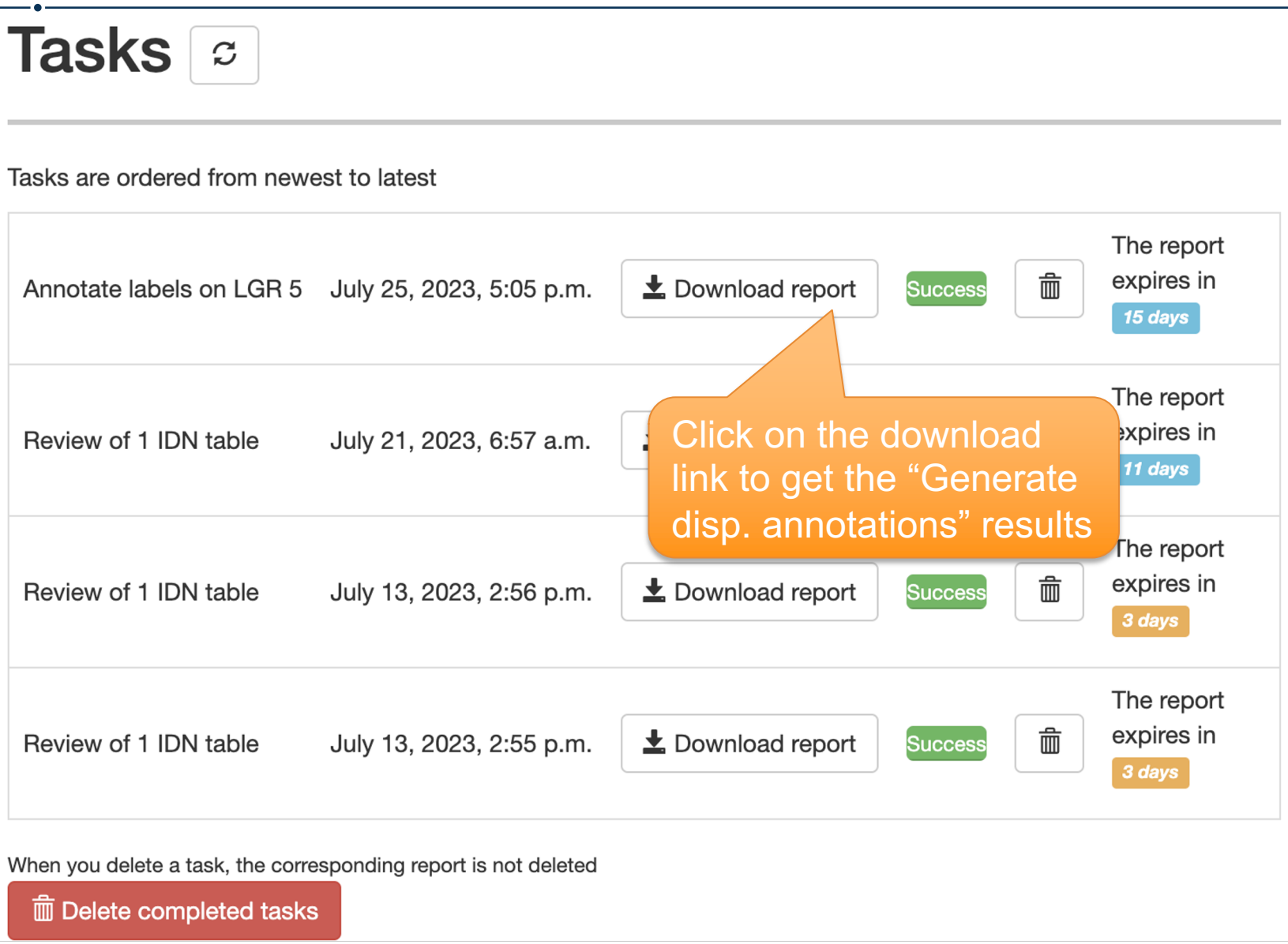
Enter Details for Single LGR



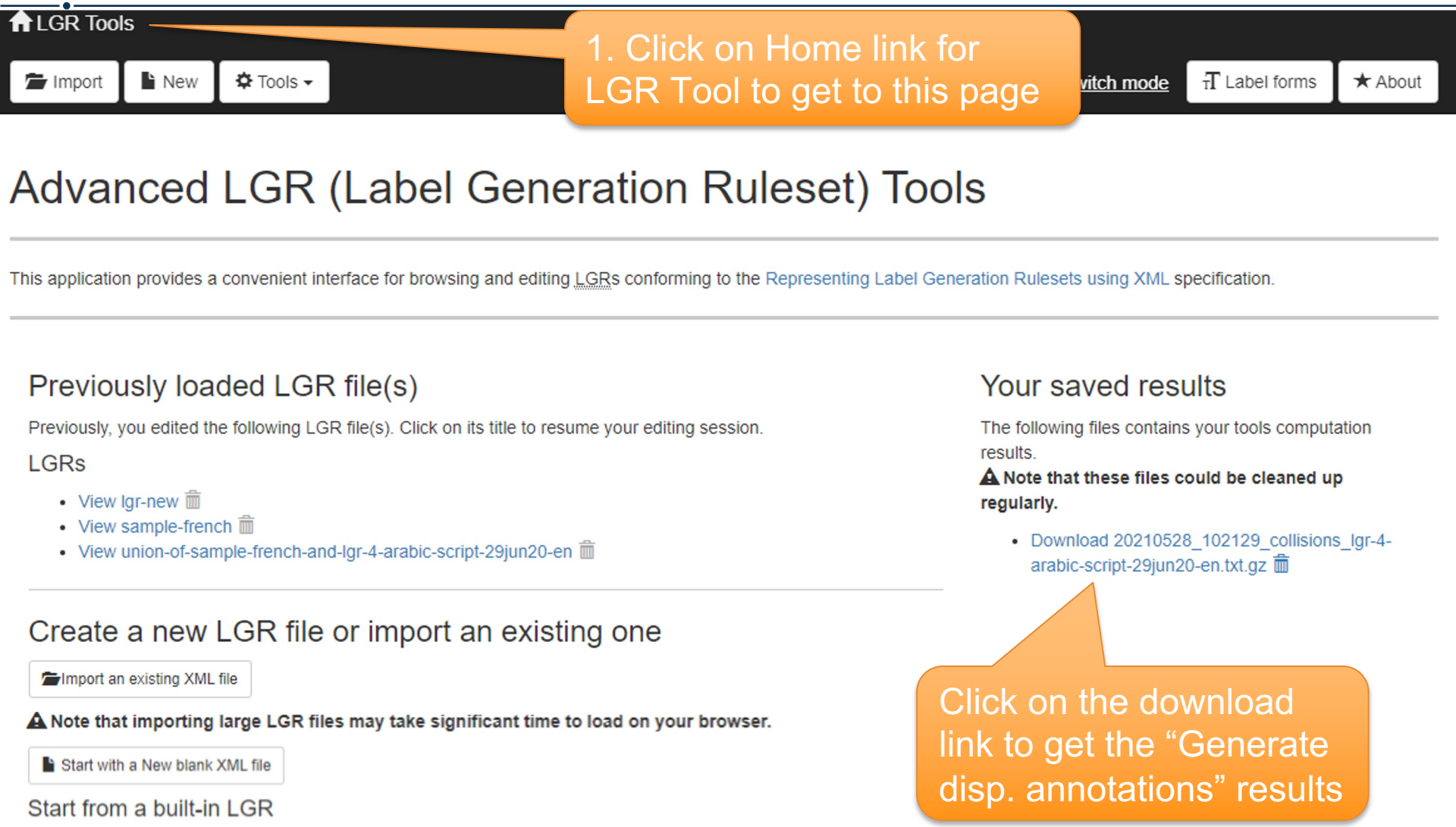
Enter Details for Merged LGR



Download Results from Task Status Page



Download Results



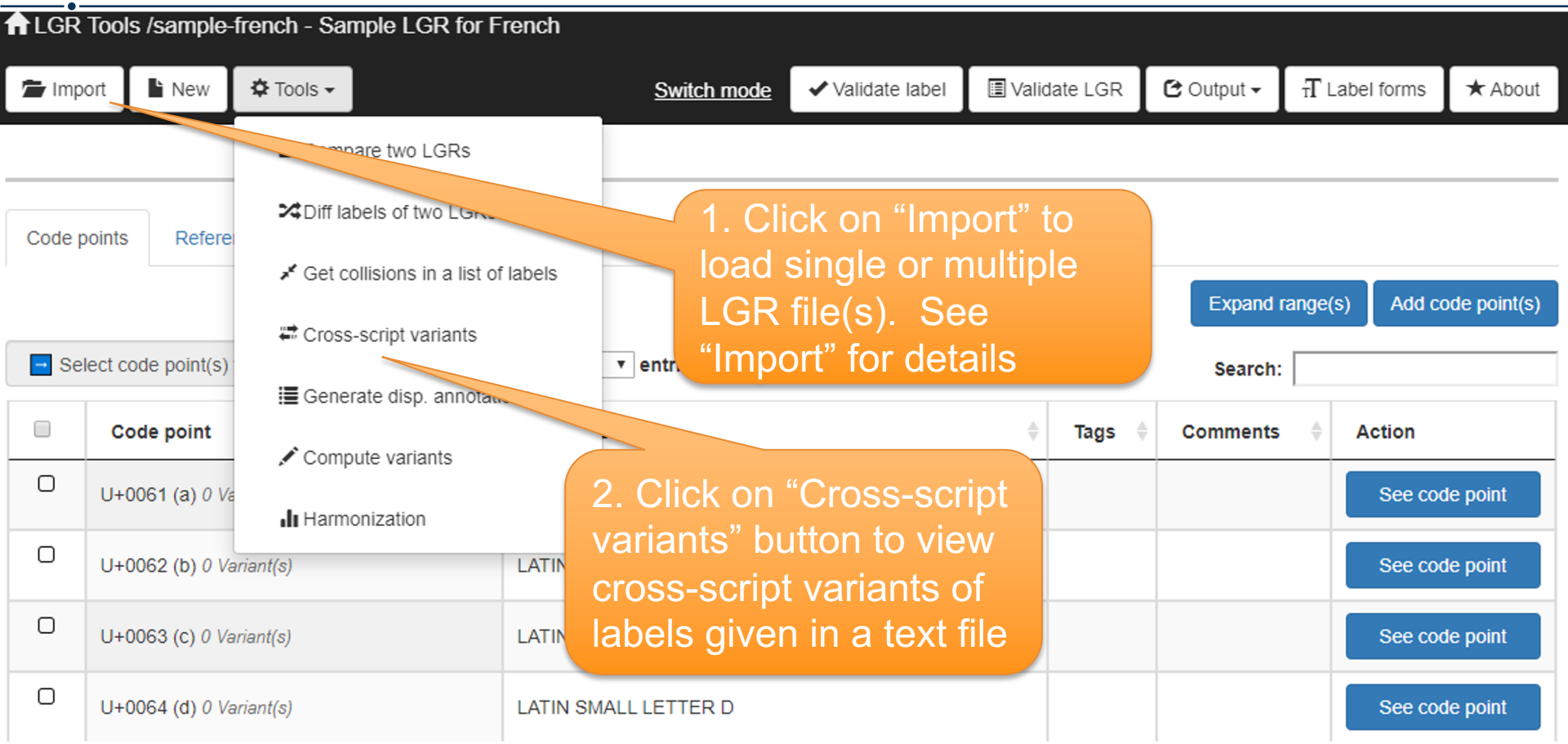
Results

```
ຫ້ຽໄຕ: valid
ໄປນອກ: valid
ເຮືອຈັກ: valid
ຮົ່ມສີແດງ: valid
ຈ່ຳເຈ: invalid
 Code point U+0EB3 not in repertoire
ຫຸ່ມແພງ: valid
ອະນຸດຕະຣະ: valid
ສາຍເລືອດແດງ: valid
ສິ່ນເຫດ: valid
ລຸກແຕ່ເຊົ້າ: valid
ສຸຂະ: valid
ຣະຈິດ: valid
ເຮັດບໍ່ໄຫວ: valid
ນັກຟິຊິກ: valid
ດ້ຽງດາດ: valid
ລະເຫລີ່ງ: valid
ເຢັນຕາຮ້ອກ: 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
ພະຜູ້ເຢັນເຈົ້າ: valid
ຫມ້ຽ: valid
ທຸກຫົ່ນທຸກແຫ່ງ: valid
ແມ່ສີ: valid
ງ໊ວນ້ອຍ: valid
ຫລອກລໍ: valid
ຕາມຄວນ: valid
ສຸພາບອ່ອນຫວານ: valid
ບໍ່ຫົ່ນ: valid
ຫມໍ: valid
ແຮງຄົນ: valid
```

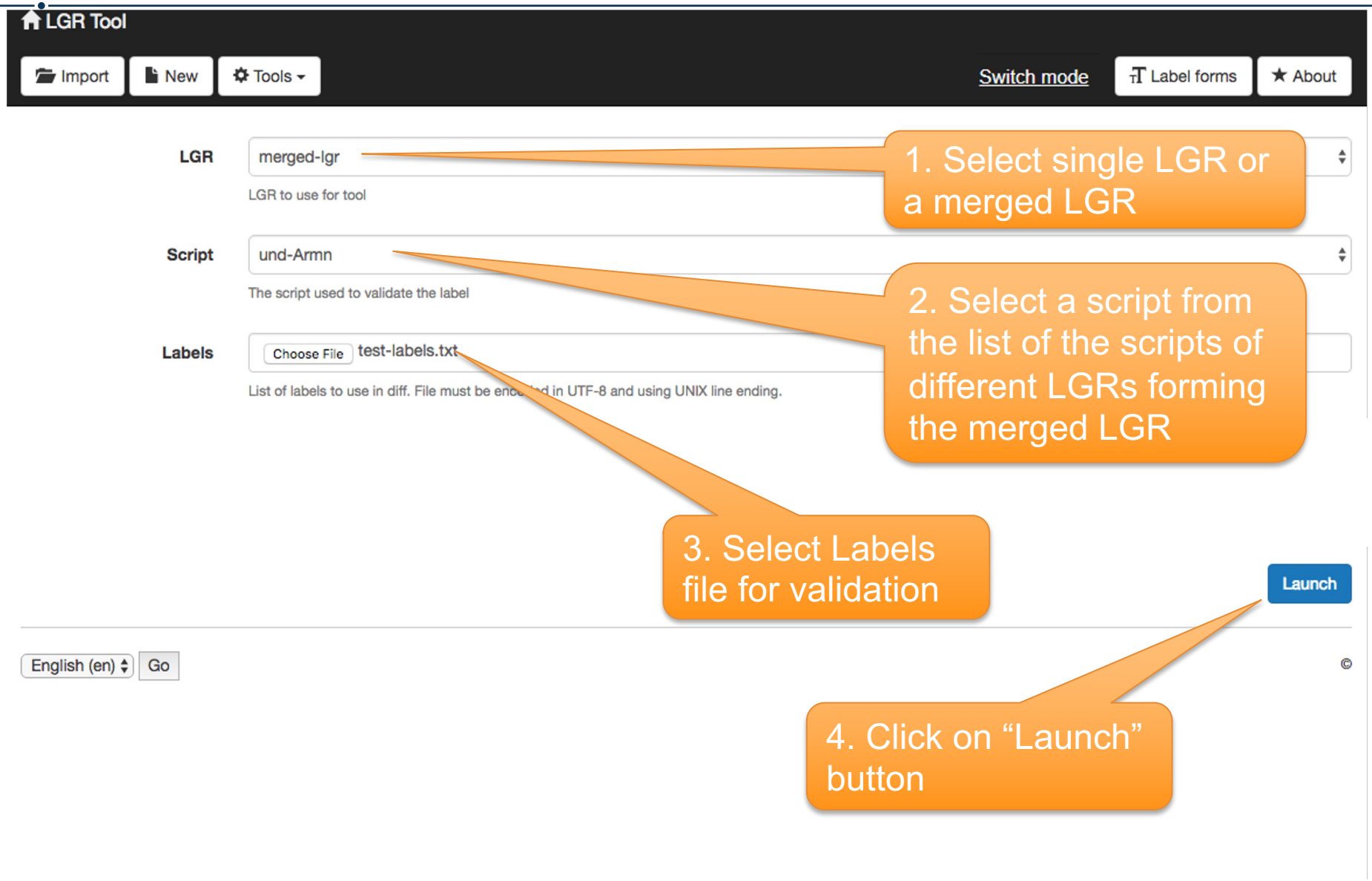
Validation result of each label is written next to it. For invalid labels, the tool also specifies the reason/rule due to which the label is invalid

View Cross-Script Variants of Labels

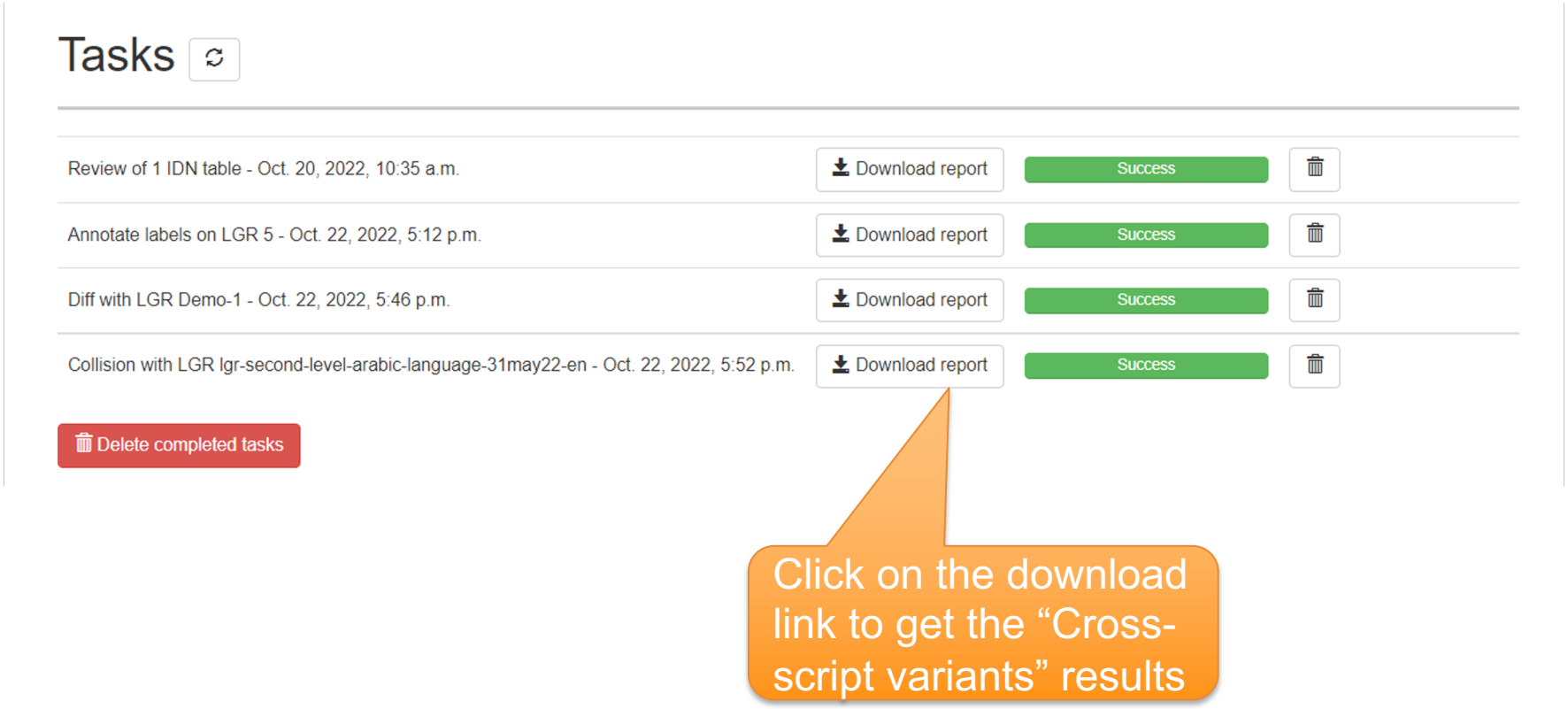
Select Cross-script Variants



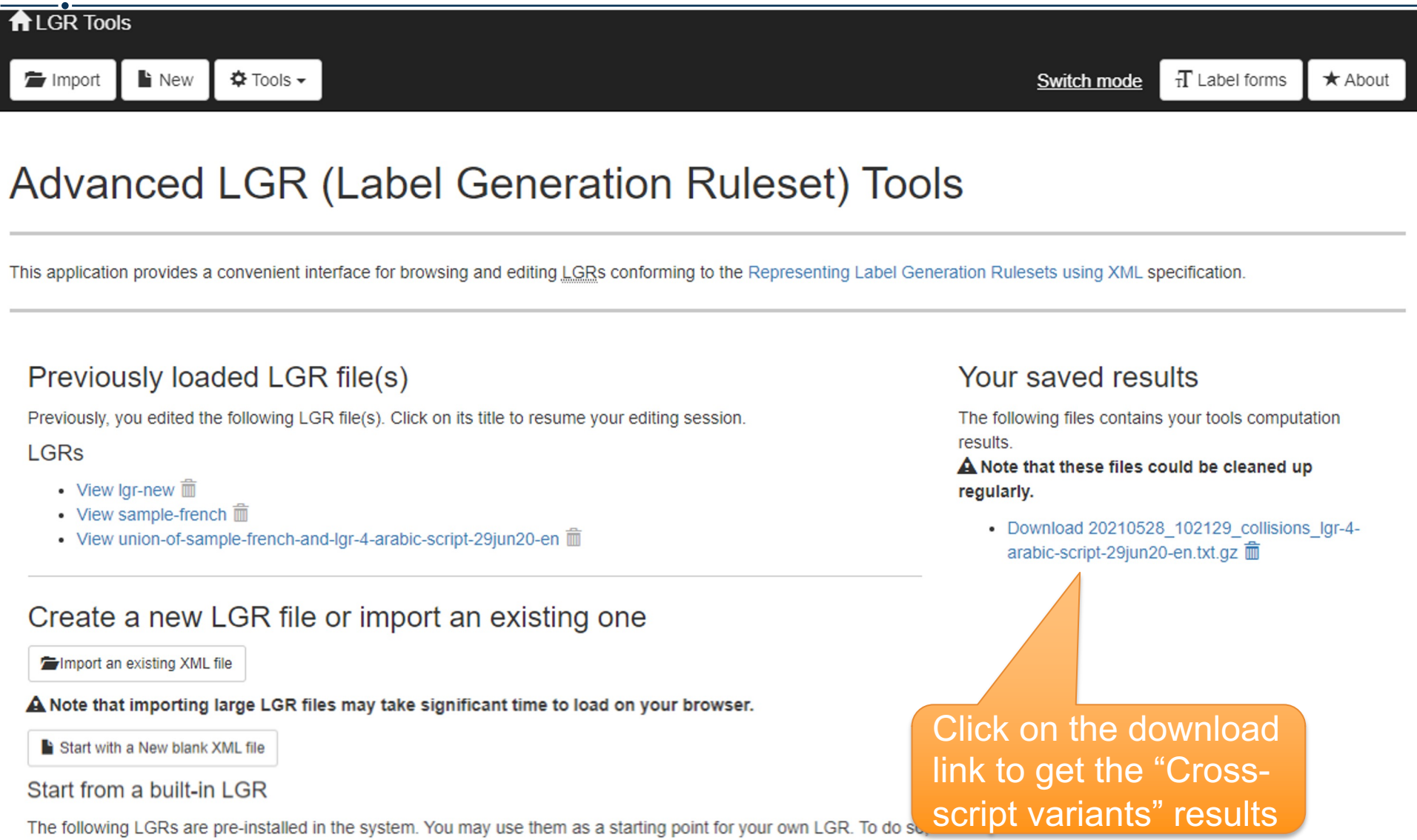
Enter Details



Download Results from Task Status Page



Download Results



Results

```
Input label U+0561 U+0575 U+0580 (այր) has cross-script variants:
       - Cross-variant U+0448 U+0575 U+0580 (шյр), disposition invalid:
              + U+0448 (ш): Cyrl
Input label U+0581 U+0561 U+0579 U+0586 (ցաչֆ) has cross-script variants:
       - Cross-variant U+0581 U+0448 U+0579 U+0586 (ցшչֆ), disposition invalid:
              + U+0448 (ш): Cyrl
       - Cross-variant U+0067 U+0561 U+0579 U+0586 (gաչֆ), disposition invalid:
              + U+0067 (g): Latn
       - Cross-variant U+0067 U+0448 U+0579 U+0586 (gшչֆ), disposition inv
              + U+0448 (ш): Cyrl
              + U+0067 (g): Latn
Input label U+0566 U+0578 U+057D (զոս) has cross-script variants:
       - Cross-variant U+0566 U+0578 U+0075 (զոu), disposition invalid:
              + U+0075 (u): Latn
       - Cross-variant U+0566 U+006E U+057D (զnս), disposition invalid:
              + U+006E (n): Latn
       - Cross-variant U+0566 U+006E U+0075 (զnu), disposition invalid:
              + U+0075 (u): Latn
              + U+006E (n): Latn
       - Cross-variant U+0071 U+0578 U+057D (qոս), disposition invalid:
              + U+0071 (q): Latn
       - Cross-variant U+0071 U+0578 U+0075 (qոu), disposition invalid:
              + U+0071 (q): Latn
              + U+0075 (u): Latn
       - Cross-variant U+0071 U+006E U+057D (qnս), 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
```

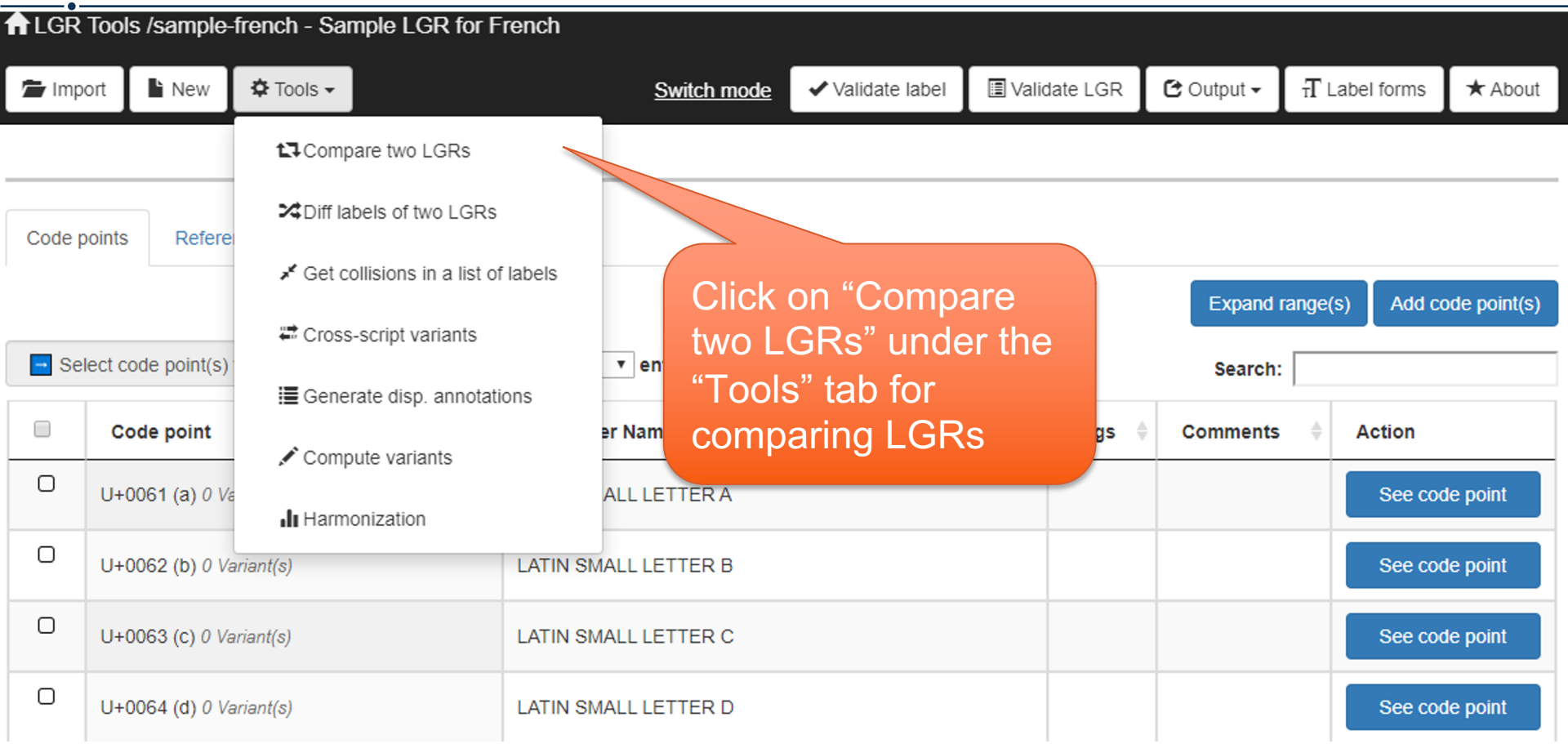
Specifies the LGR (from a set of multiple LGRs) that is used to generate these cross-script variants

Cross-script variants of the label

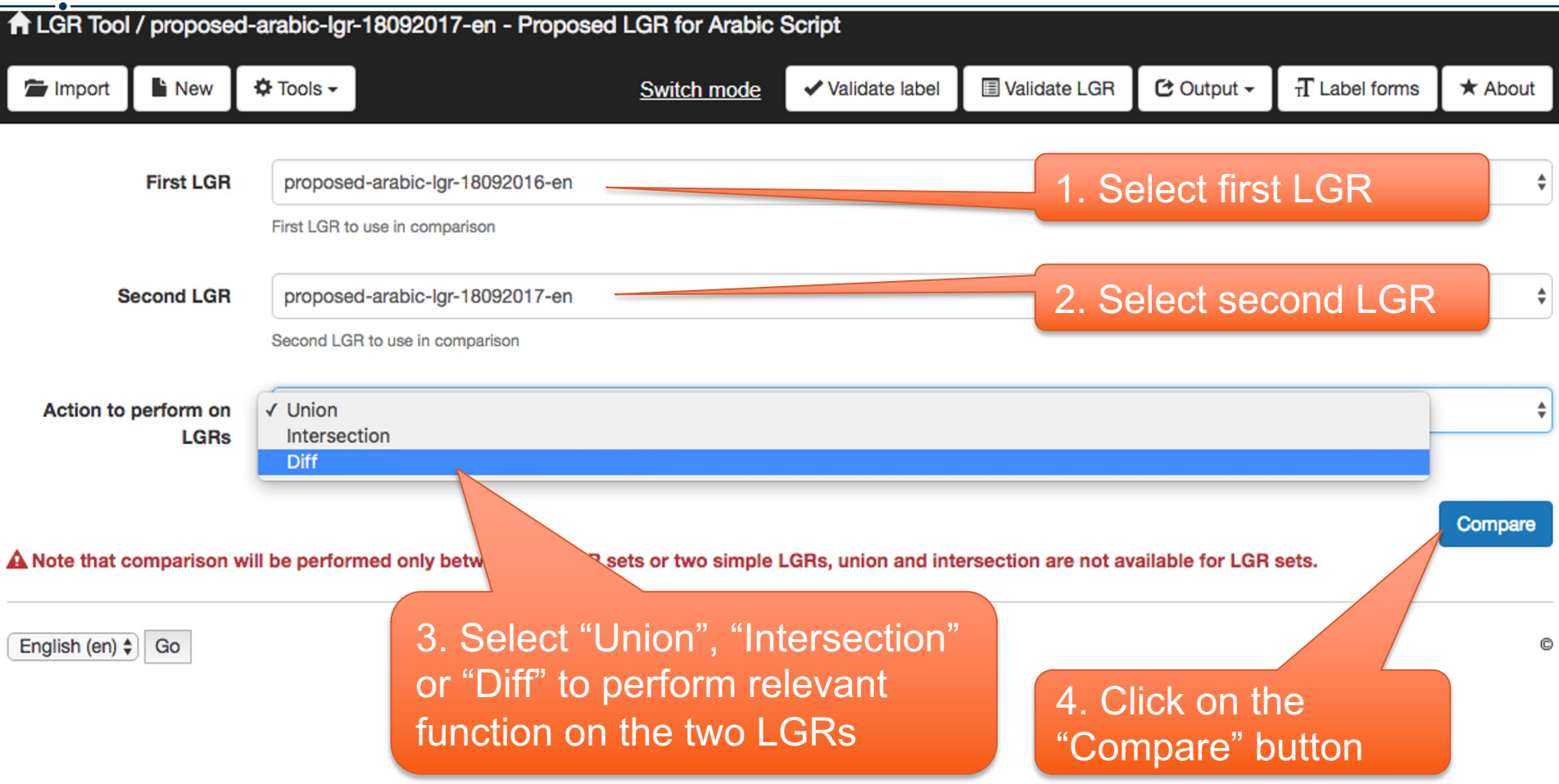
Disposition of cross-script variants

Compare LGRs

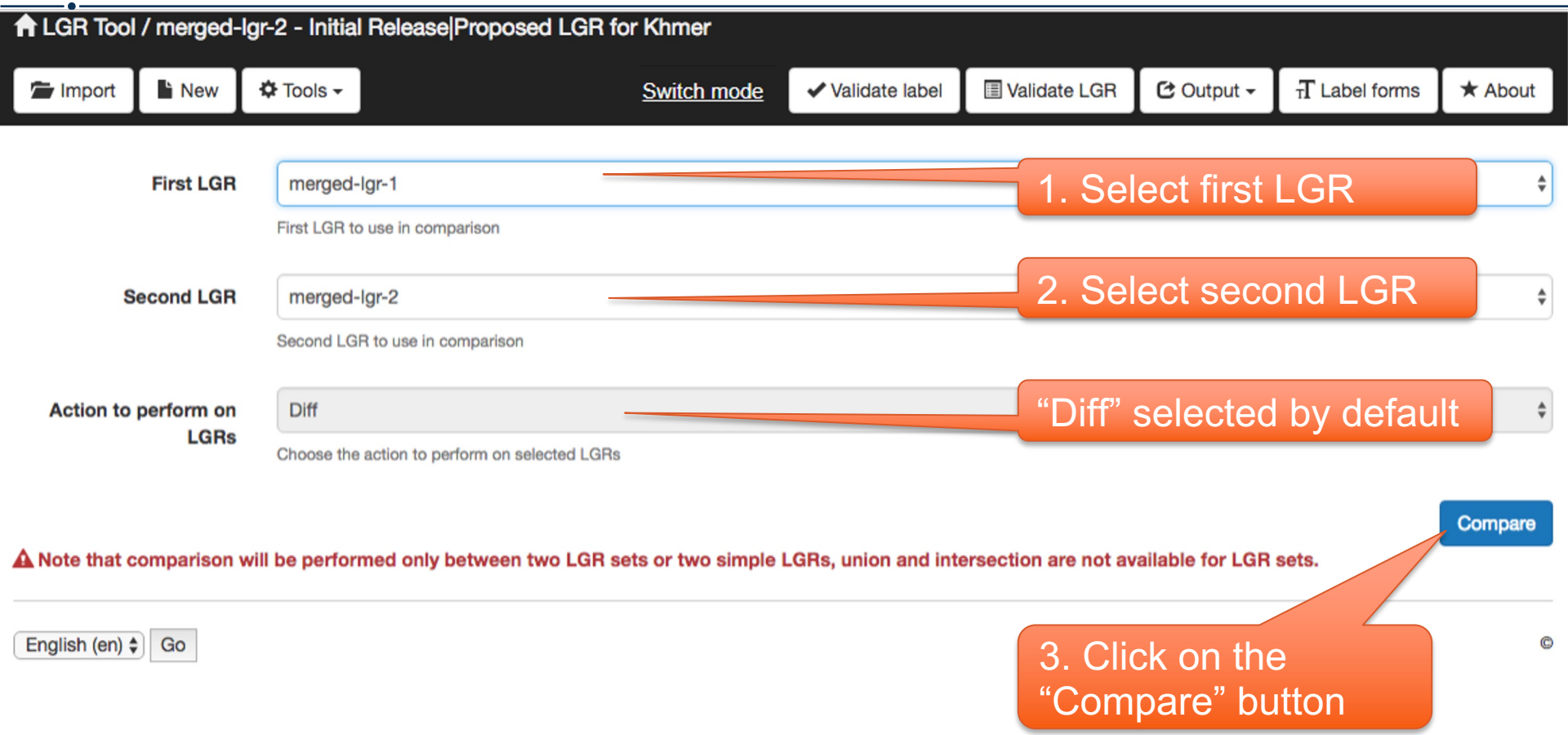
Compare LGRs



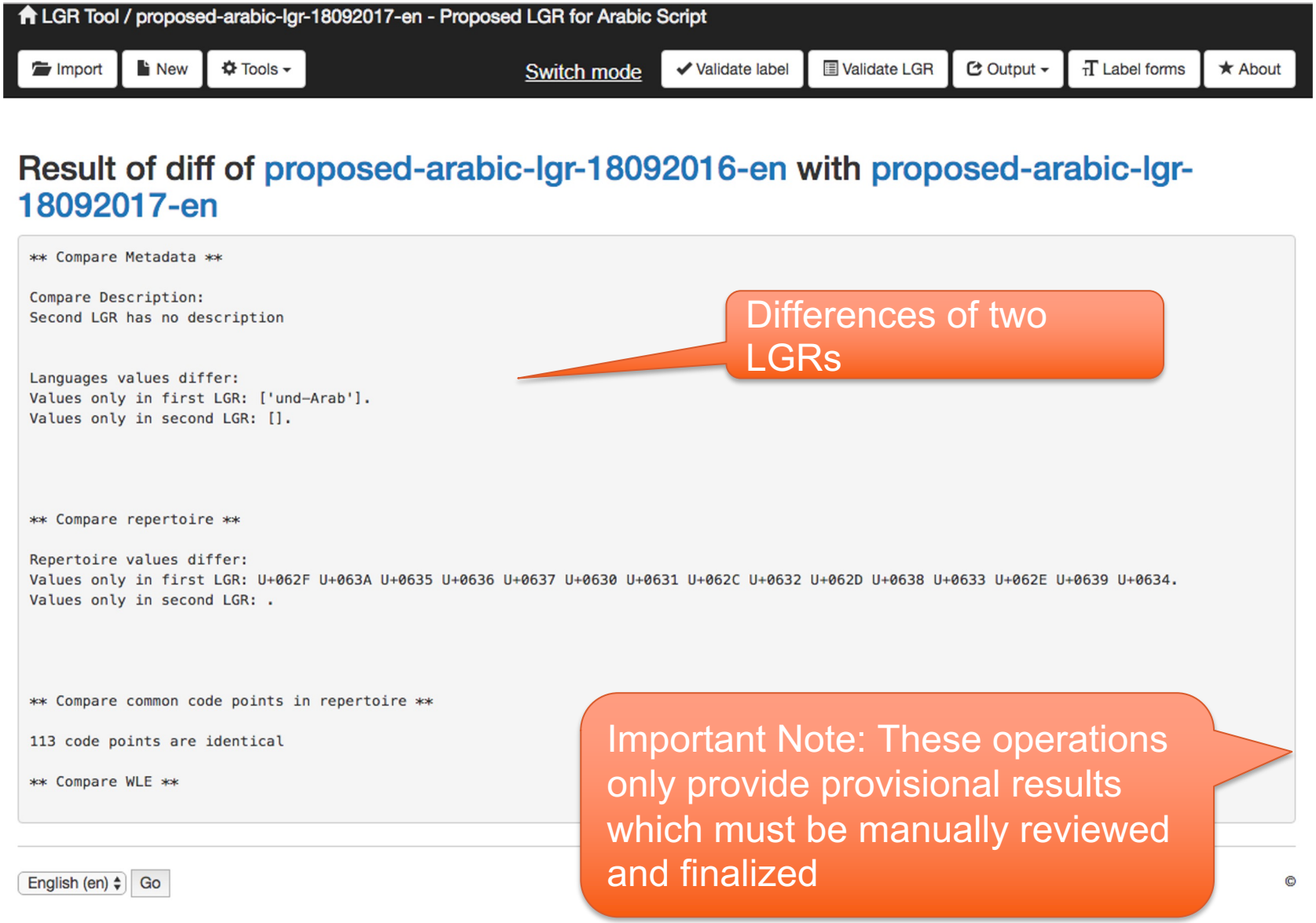
Union, Intersection or Difference



With Merged LGRs, Only Difference



Result of Difference function

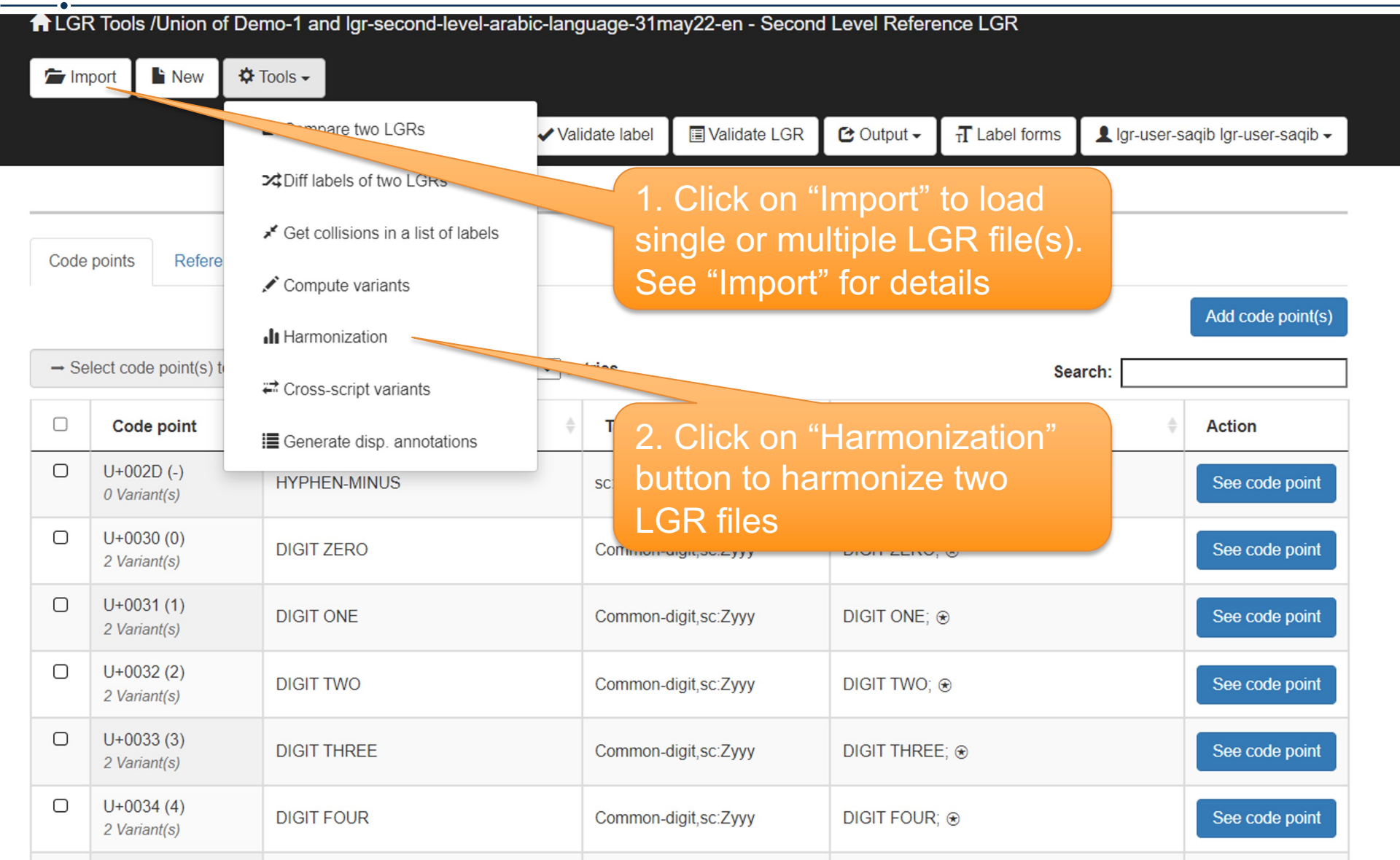


Differences of two LGRs

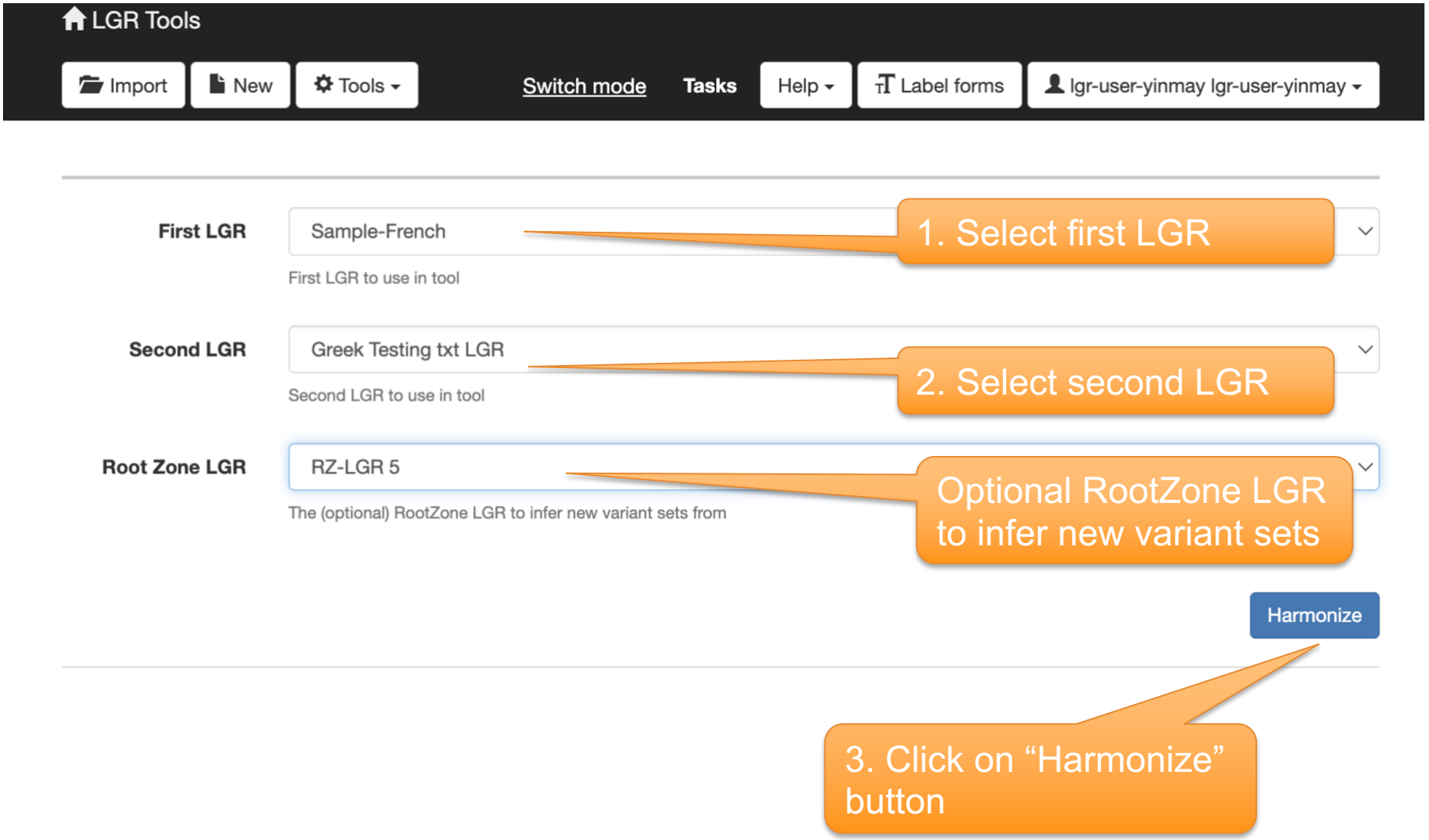
Important Note: These operations only provide provisional results which must be manually reviewed and finalized

Harmonize Multiple LGRs

Select Harmonization



Enter Details

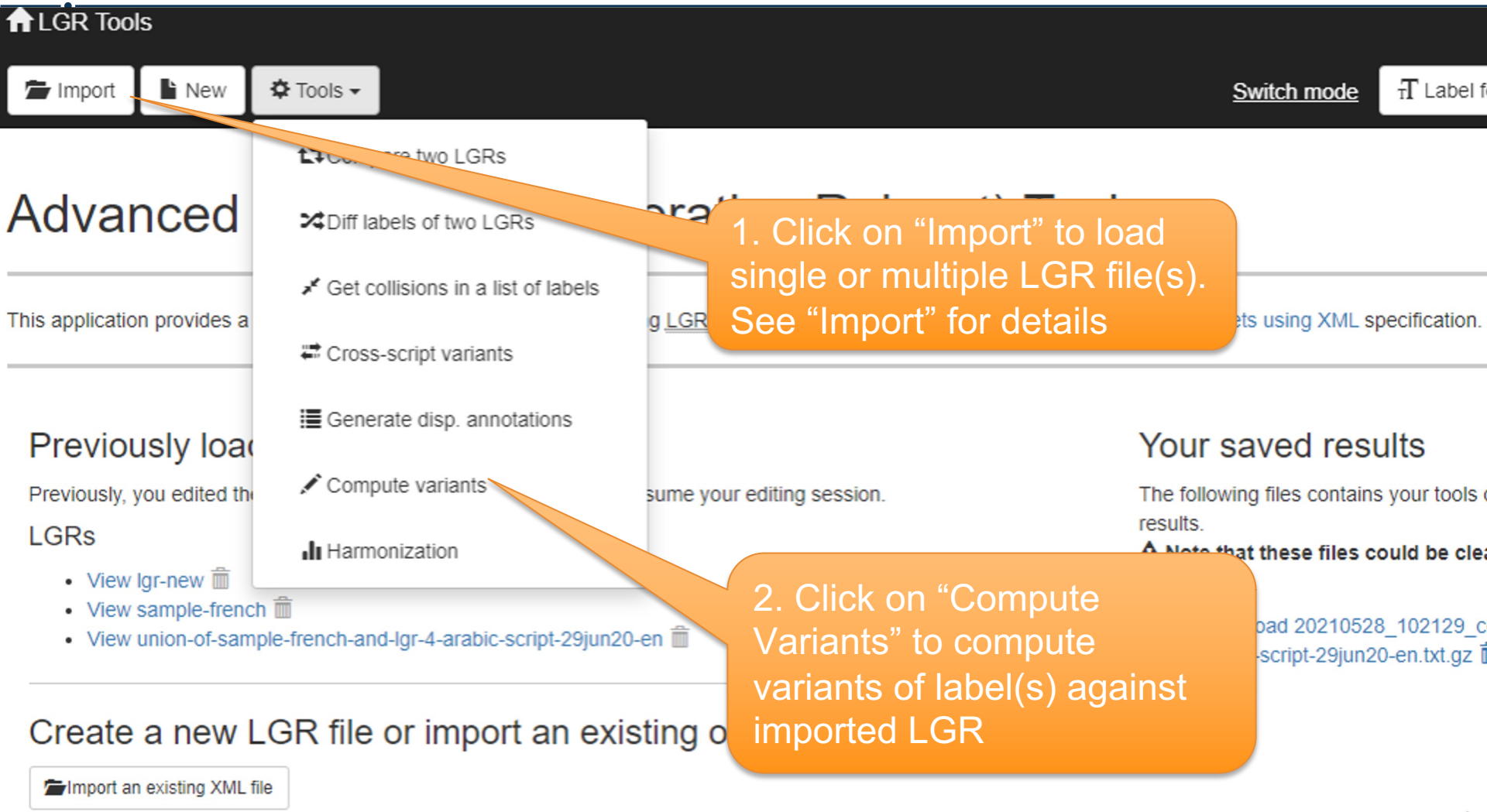


Results

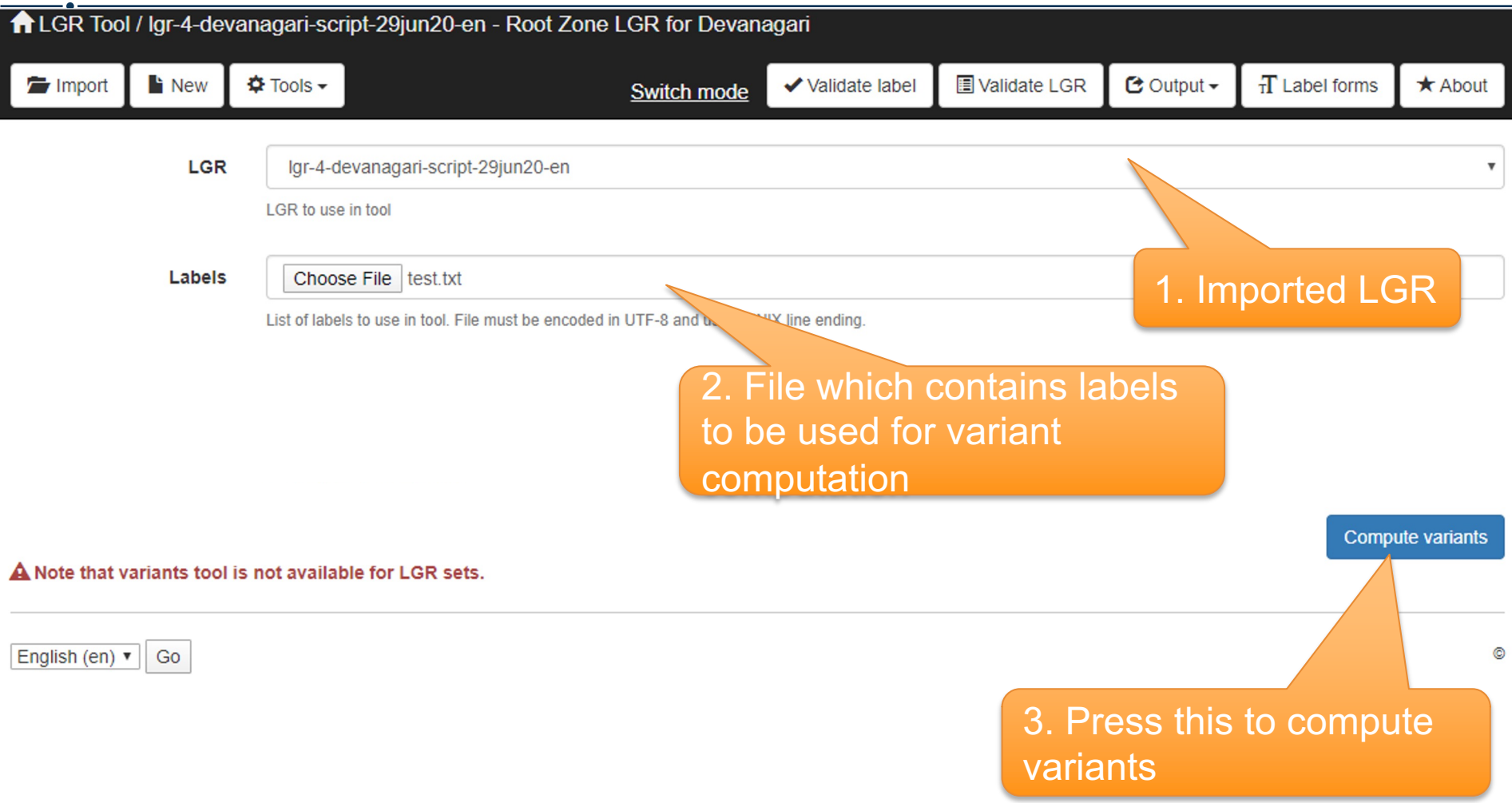
The Harmonized LGRs will also be listed on the homepage

Compute Variants

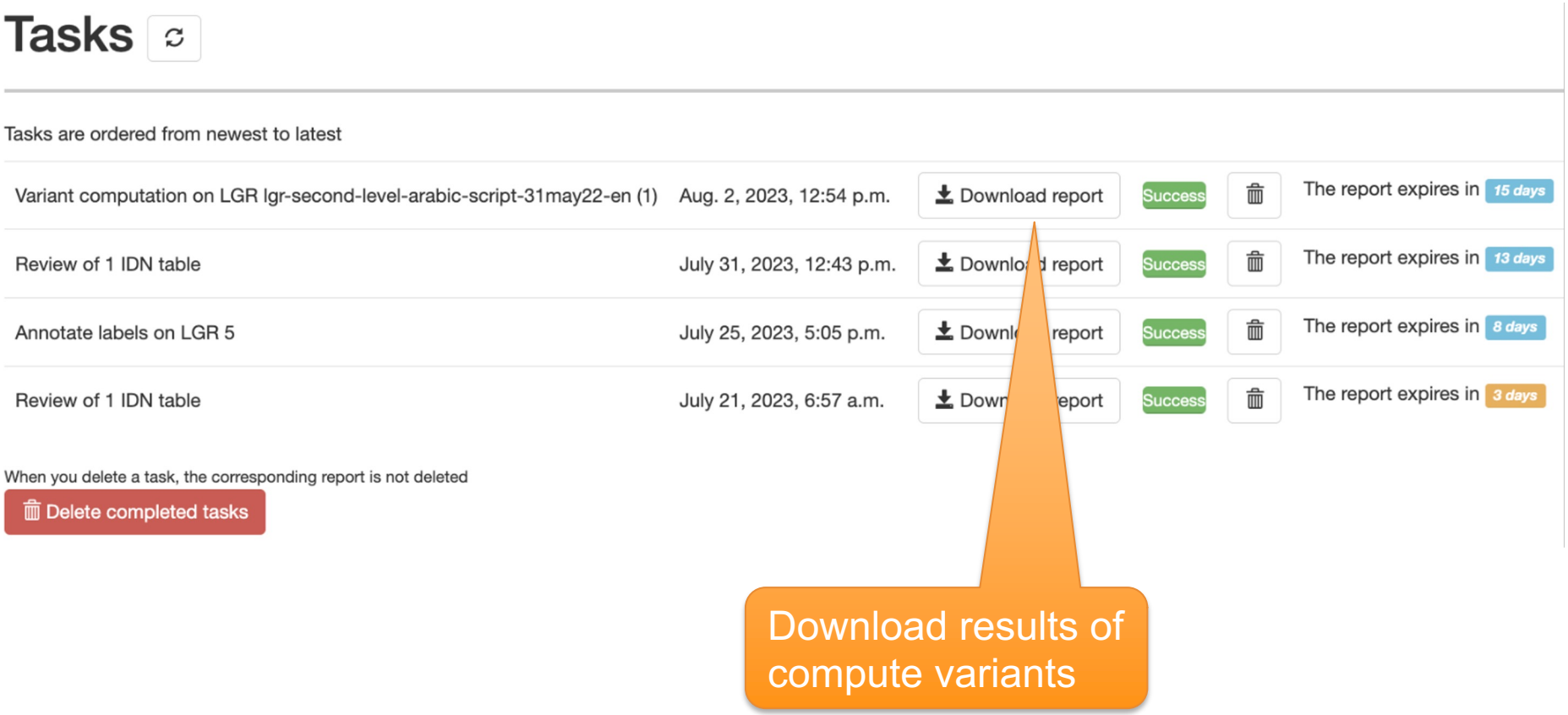
Compute Variants



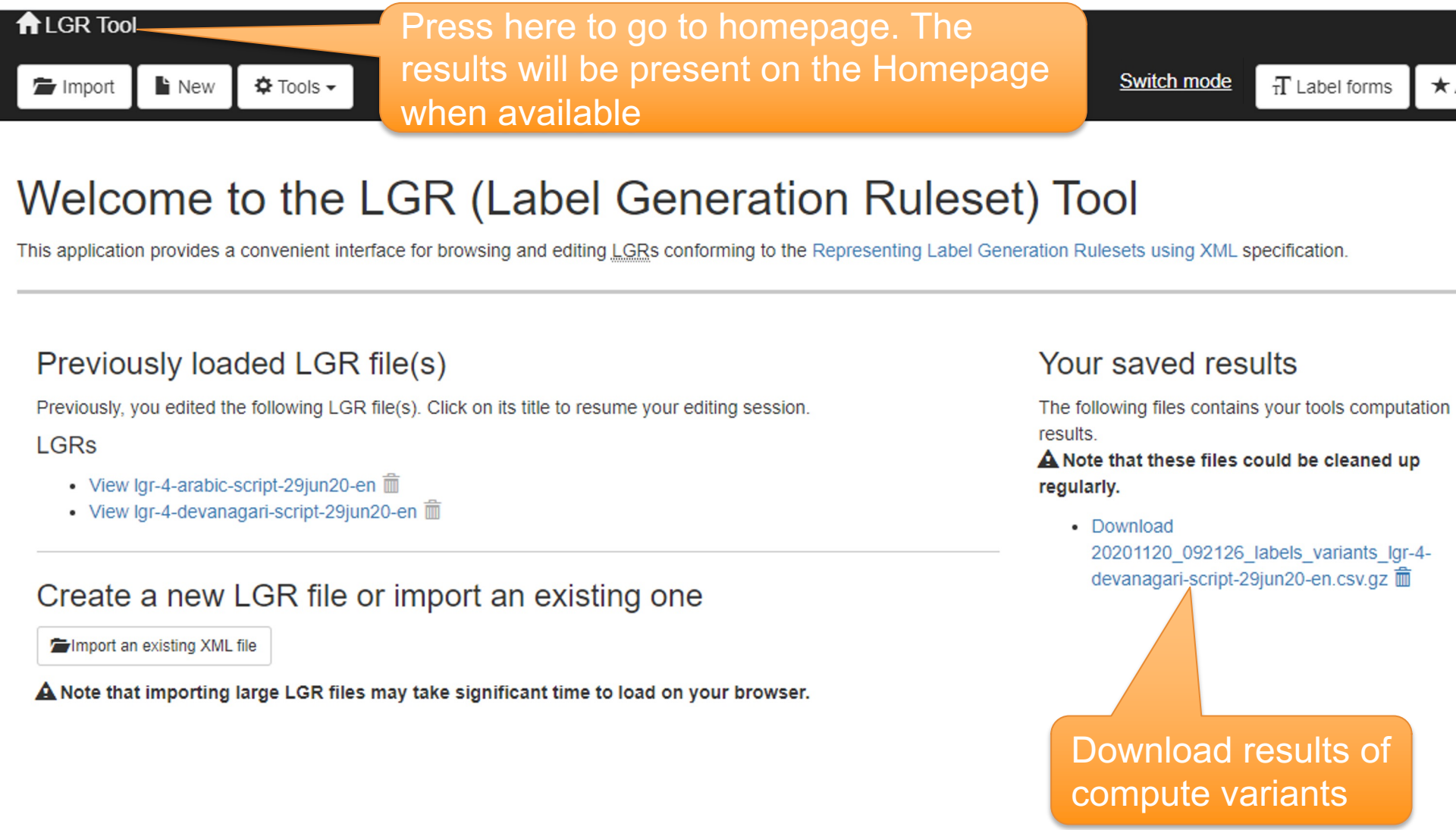
Enter Details



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.
Success
The report expires in 13 days
Annotate labels on LGR 5
July 25, 2023, 5:05 p.m.
Success
The report expires in 8 days
Review of 1 IDN table
July 21, 2023, 6:57 a.m.
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

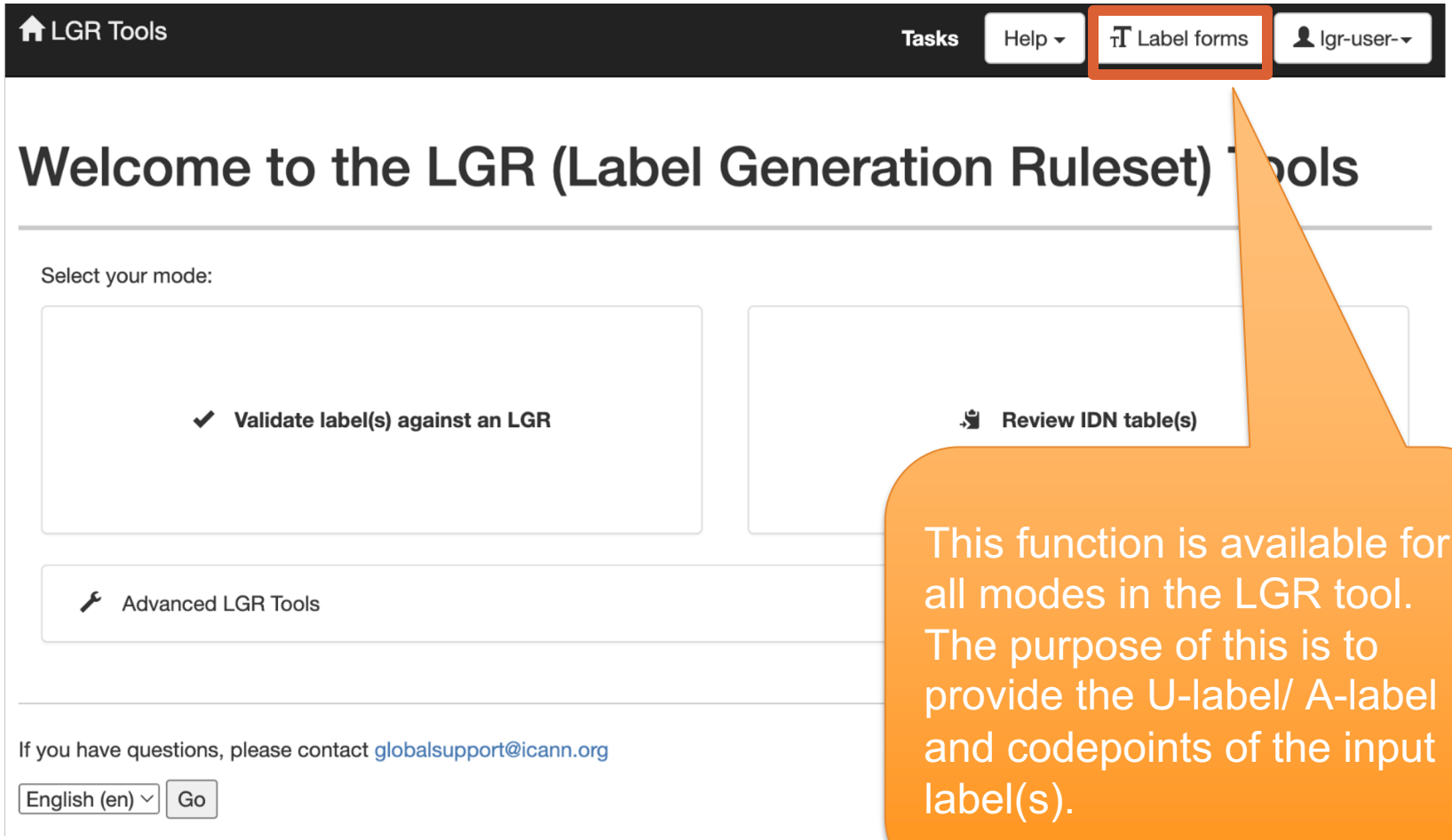

LGR Tool
Import
New
Tools
Switch mode
Label forms
Press here to go to homepage. The results will be present on the Homepage when available
Welcome to the LGR (Label Generation Ruleset) Tool
This application provides a convenient interface for browsing and editing LGRs conforming to the Representing Label 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

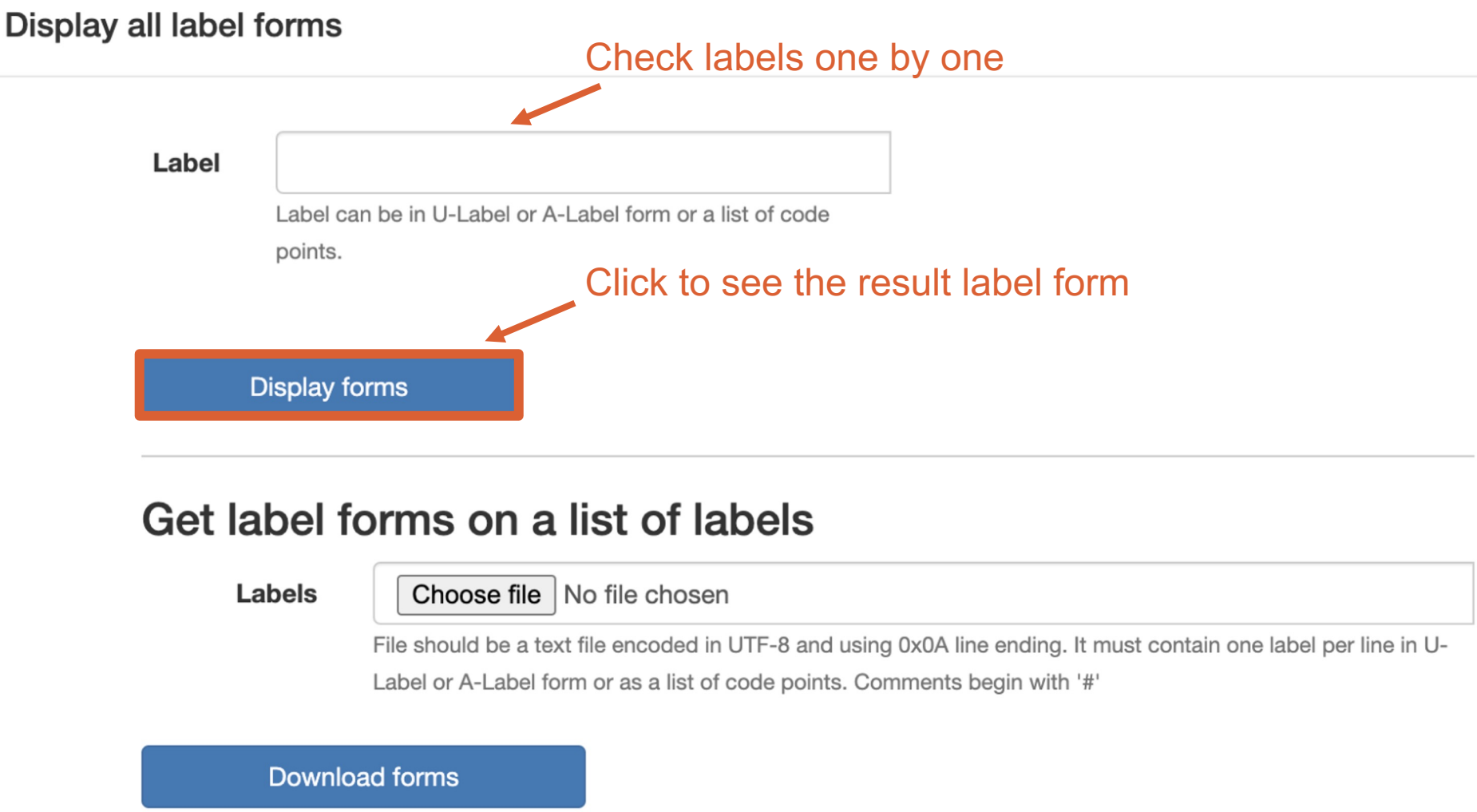
A label, U label against IDNA2008

Label Forms

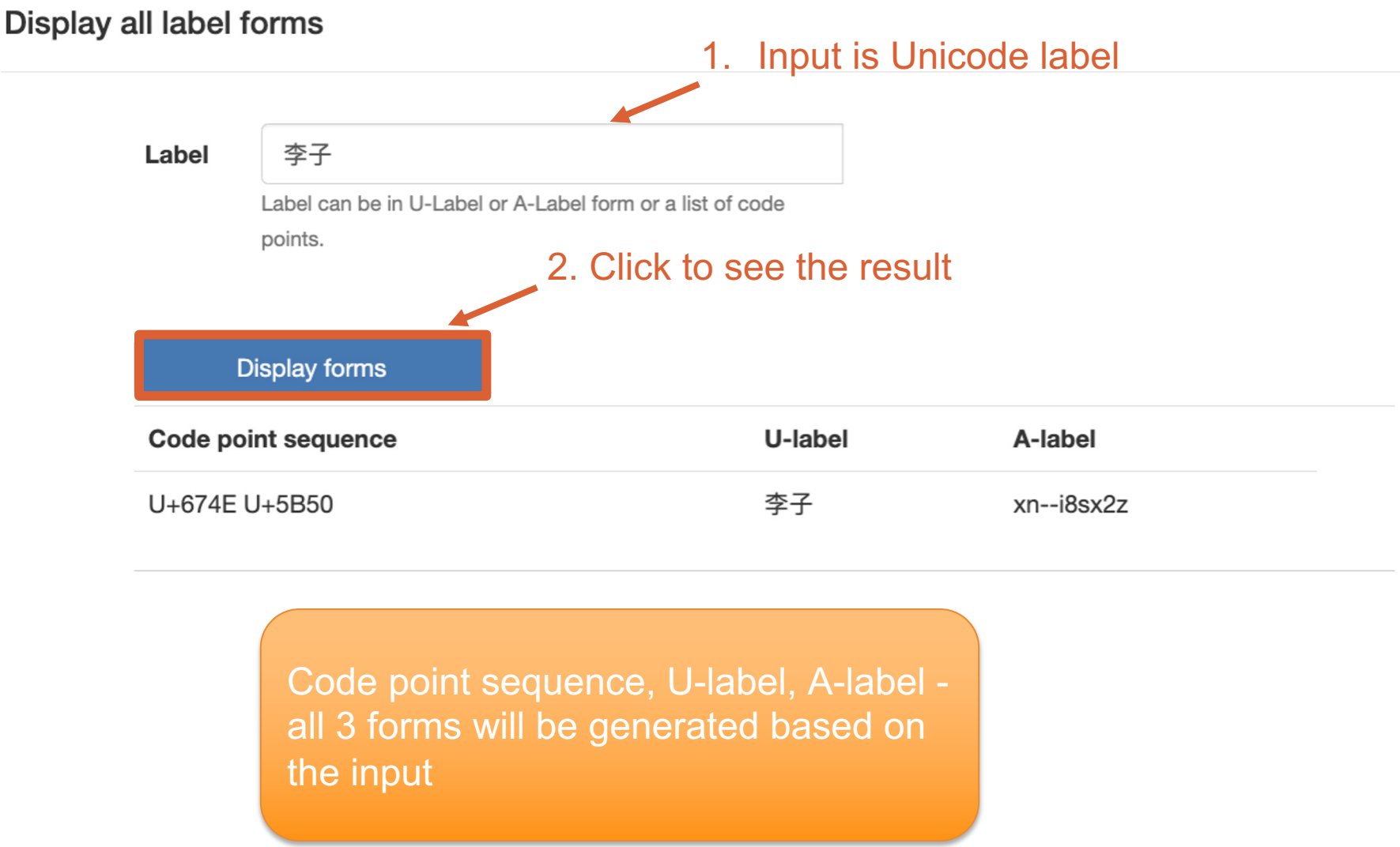
- Type in a label or upload a list of labels



Label Forms

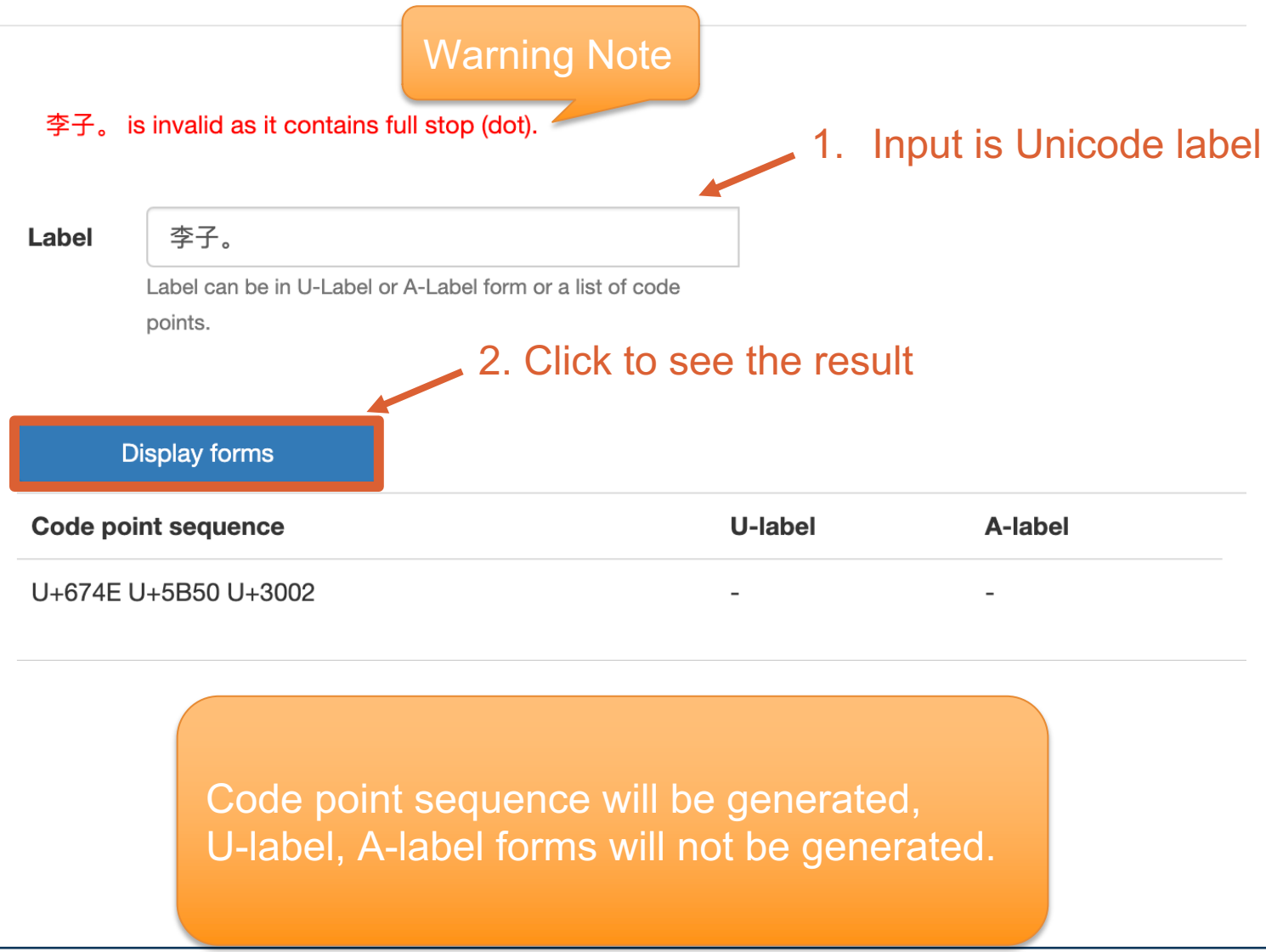

Display all label forms
Check labels one by one
Label
Label can be in U-Label or A-Label form or a list of code points.
Click to see the result label form
Display forms
Get label forms on a list of labels
Labels
Choose file
No file chosen
File should be a text file encoded in UTF-8 and using 0x0A line 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)



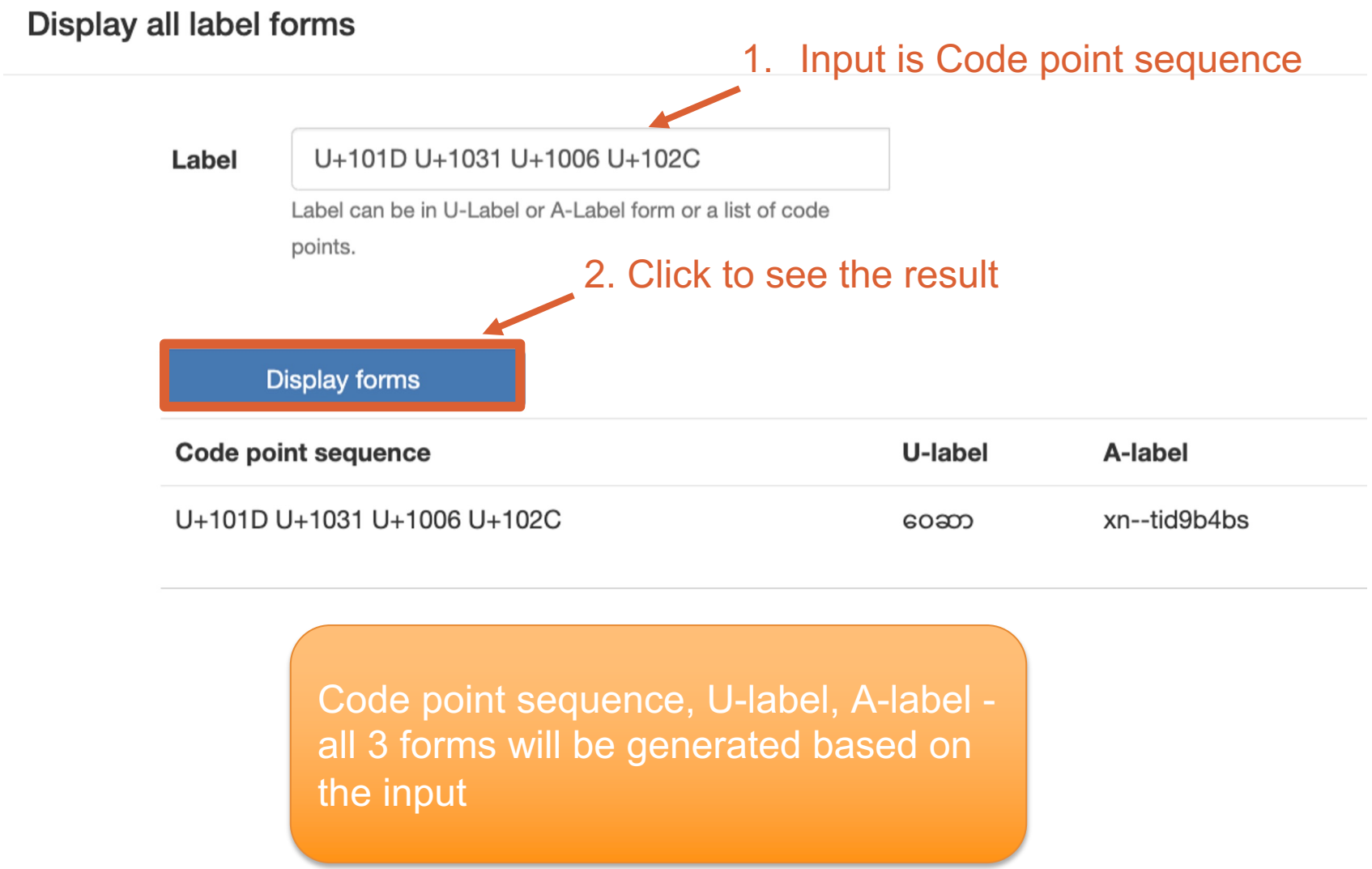
Code point sequence, U-label, A-label - all 3 forms will be generated based on the input

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



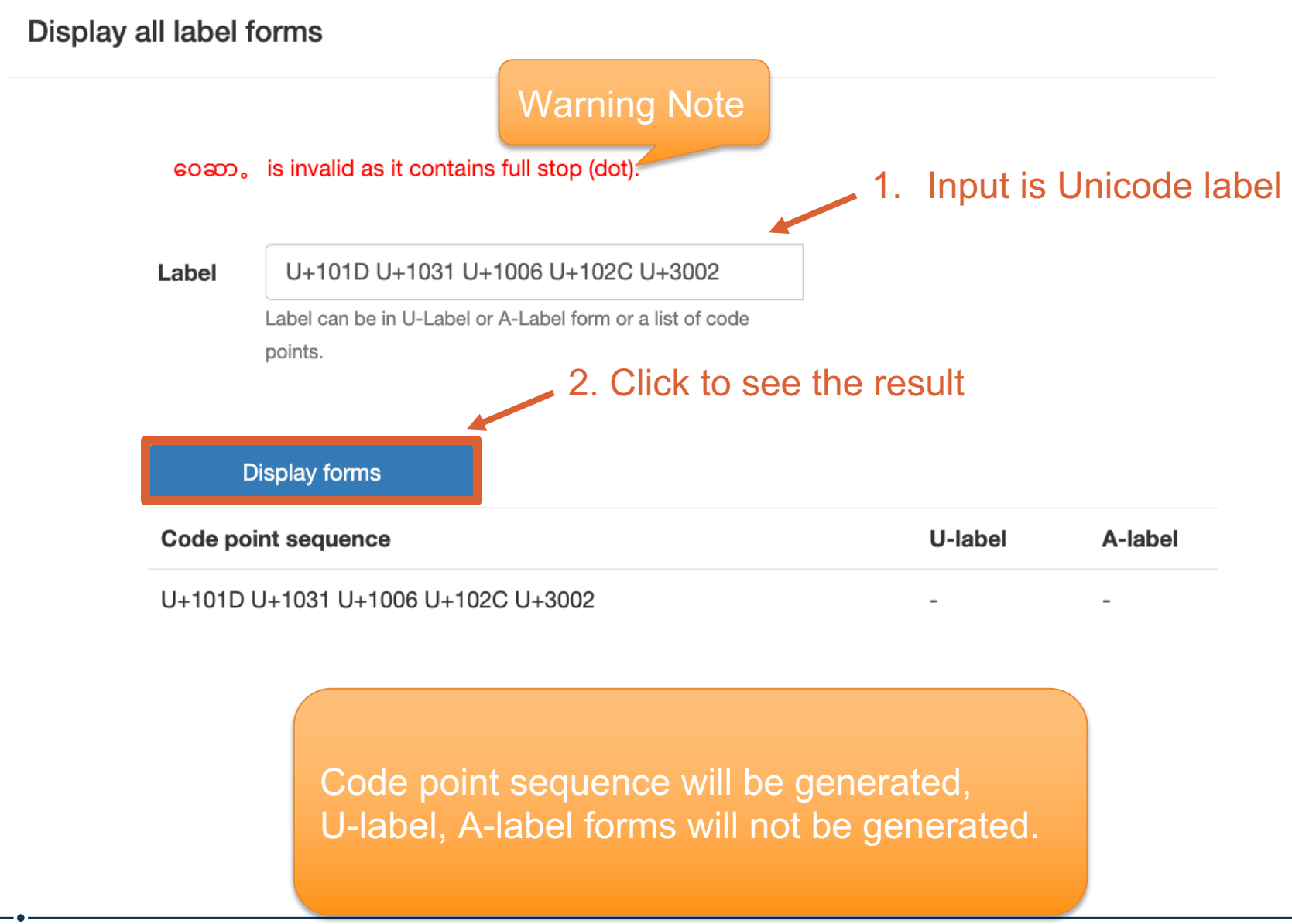
Code point sequence will be generated,
U-label, A-label forms will not be generated.

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

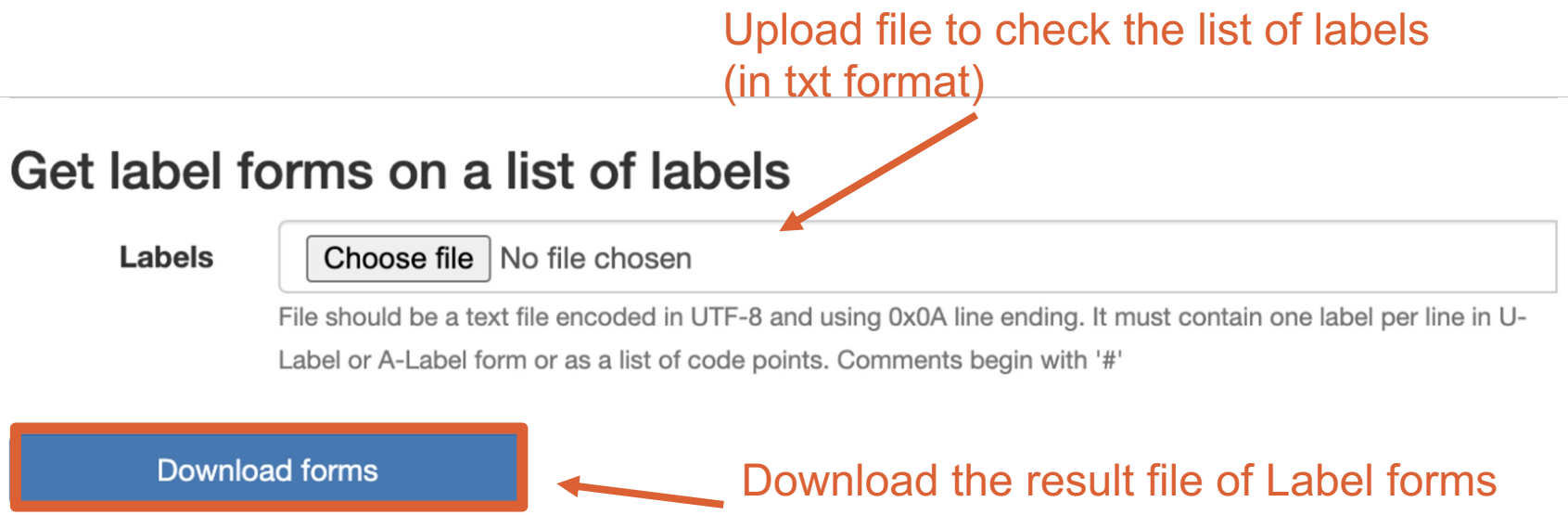


Code point sequence, U-label, A-label - all 3 forms will be generated based on the input

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



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



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	စိန်	xn--sid4a9d7b	-
U+1005 U+102D U+1014 U+103A U+002D	U+1005 U+102D U+1014 U+103A U+002D	-	-	စိန်- 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	စိန်	xn--sid4a9d7b	-
U+1005 U+102D U+1014 U+103A U+002D	U+1005 U+102D U+1014 U+103A U+002D	-	-	စိန်- 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


facebook.com/icannorg


weibo.com/ICANNorg


linkedin.com/company/icann


flickr.com/photos/icann


youtube.com/user/icannnews


slideshare.net/icannpresentations