Transliteration of Tamil and Other Indic Scripts

Ram Viswanadha
Unicode Software Engineer
IBM Globalization Center of Competency, California, USA

Main points of Powerpoint presentation

This talk gives an overview and discusses the issues with transliteration of Indic scripts to Latin and between different Indic scripts, e.g: Tamil-Telugu, Gujarati-Tamil, etc.

It is often perceived that transliteration between different Indic scripts is straightforward because all Indic scripts have a common origin in Brahmi script.

The ISCII standard is based on this similarity between scripts, and the placement of Unicode code points for each Indic script is based on an early version of ISCII. Correct transliteration between Indic scripts takes advantage of this allocation but handles special cases. The ICU implementation uses a script-neutral pivot range in Unicode.

Agenda

- What is ICU?
- Terminology & Concepts
- Standards for Romanization
- Problems in Romanization
- Problems in Inter-Indic Transliteration
- Implementation approaches
- Implementation in ICU
- Summary

•A brief overview of what International Components for Unicode (ICU) is.
•Some terms and concepts which are important for the scope of this presentation are discussed
•Different standards for Romanization, ISCII and ISO 15919 in particular are presented
•Some problems in Romanization and Inter-Indic transliteration are discussed
•Different implementation approaches, their deficiencies and how ICU's implementation tries to solve them are presented
-----

What is ICU?

* Internationalization libraries for C, C++, Java*
  – Open source – non-viral
  – Sponsored by IBM
* Sun's Java licenses an earlier ICU version; ICU4J updates it.
* Unicode standard compliant

– full supplementary support

* Cross-platform; extensible and customizable

* High performance and thread-safe

– Multiple locales in same thread – simultaneously

* http://oss.software.ibm.com/icu/

ICU (International Components for Unicode) is a collaborative, open-source development project jointly managed by a group of companies and individual volunteers throughout the world, using the Internet and the Web to communicate, plan, and develop the software and documentation. It is sponsored and used by IBM.

Comprehensive support for the Unicode Standard is the basis for multilingual, single-binary software. ICU uses the most current versions of the standard, and provides full support for supplementary characters.

As computing environments become more heterogeneous, software portability becomes more important. ICU lets your produce the same results across all the various platforms you support. It offers great flexibility to extend and customize the supplied system services.

For more information, see the ICU website.

Terminology

- Transformation
- Script Transliteration / Transliteration
- Translation
- Diacritics
- Romanization

Transformation is the process of converting characters from one form to another.

Transforms in ICU provide a flexible mechanism capable of handling a much broad range of tasks. In particular, Transforms have pre-built transformations for case conversions, for normalization conversions, for the removal of given characters, and also for a variety of language and script transliterations.

Script Transliteration is the process of converting characters from one script to another. For example, converting characters from Greek to Latin, or Japanese Katakana to Latin. Script Transliteration is not translation. Rather, it is the conversion of letters from one script to another without translating the underlying words or meaning.

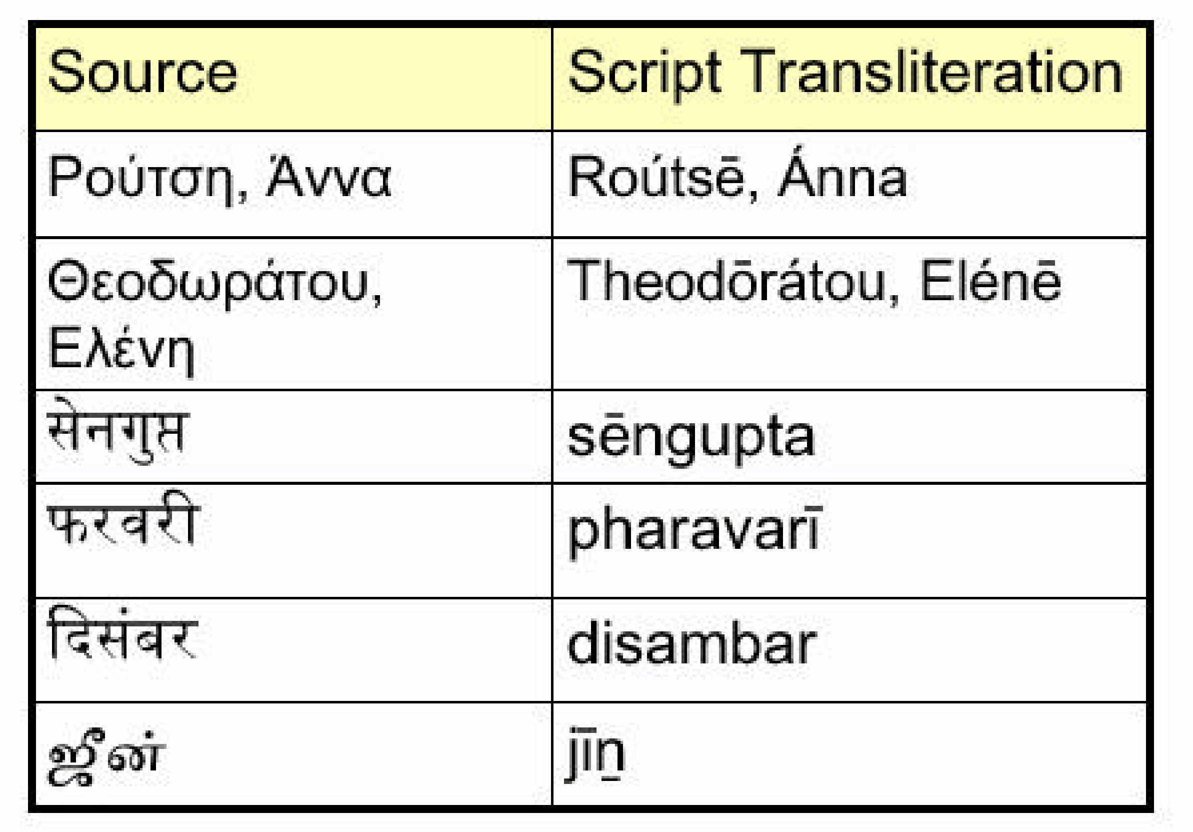
Translation: conversion of text from one language to another so that meaning can be conveyed.

Diacritics: marks added to a character to add phonetic value to a base character to distinguish words that are otherwise graphically identical, e.g: cedilla, acute accent Romanization: the process of transliterating non-Roman scripts to Latin.

Script Transliteration

Here are some examples of script-script transliterations.

For the first three rows, the values in the left column are customer names from a database. On the right are transliterations; text that will be read far more easily by the average Englishspeaking database support engineer. For the last three rows, the values in the left column are month names and on the right are transliterations.



Source	Script Transliteration
Ρούτση, Άννα	Roútsē, Ánna
Θεοδωράτου, Ελένη	Theodōrátou, Eléné
सेनगुप्त	sēngupta
फरवरी	pharavarī
दिसंबर	disambar
ஜீன்	jīṉ

Transliteration Guidelines

- Complete
- Predictable
- Pronounceable
- Unambiguous
- Partial reversibility

The following lists the general guidelines for transliterations:
•complete: every well-formed sequence of characters in the source script should transliterate to a sequence of characters in the target script.
•predictable: the letters themselves (without any knowledge of the languages written in that script) should be sufficient for the transliteration, based on a relatively small number of rules. This allows the transliteration to be performed mechanically.
•pronounceable: transliteration is not as useful if the process simply maps the characters without any regard to their pronunciation.
•unambiguous: it is always possible to recover the text in the source script from the transliteration in the target script. Someone that knows the transliteration rules will be able to recover the precise spelling of the original source text.
•partial reversibility: In script transliteration there are cases where all characters in the source script may not have one-to-one mapping for transliteration in the target script. To preserve pronunciation these characters may be mapped to some character or sequence of characters that may produce a similar sound. In such cases reversibility will be incomplete.

Standards for Romanization

- ISCII-91 : Indian Standard Code for Information Interchange
- Hunterian : Sir William Hunter's transliteration system
- ALA-LC : American Library Association – Library of Congress
- BGN/PCGN 1964 : refers to United States Board on Geographic Names and the Permanent Committee on Geographical Names for British Official Use
- ISO 15919 : International Standards Organization
- UNGEGN : United Nations Group of Experts on Geographical Names

For the purpose of discussion only ISO 15919 and ISCII are considered.

ISCII-91 :
Northern-Indic scripts:

- The implicit vowel "a" at the end of a word is eliminated
- The implicit vowel is also eliminated for nasal conjuncts where consonant is preceded by consonants
- Words ending in other conjuncts retain the implicit vowel
- Explicit Virama is required for words borrowed from Sanskrit

Sanskrit and Southern-Indic scripts:

- The implicit vowel at the end of a word is never eliminated
- If elimination is required then the transliteration is represented by a halant

Does not provide transliteration for OM symbol

ISO 15919:

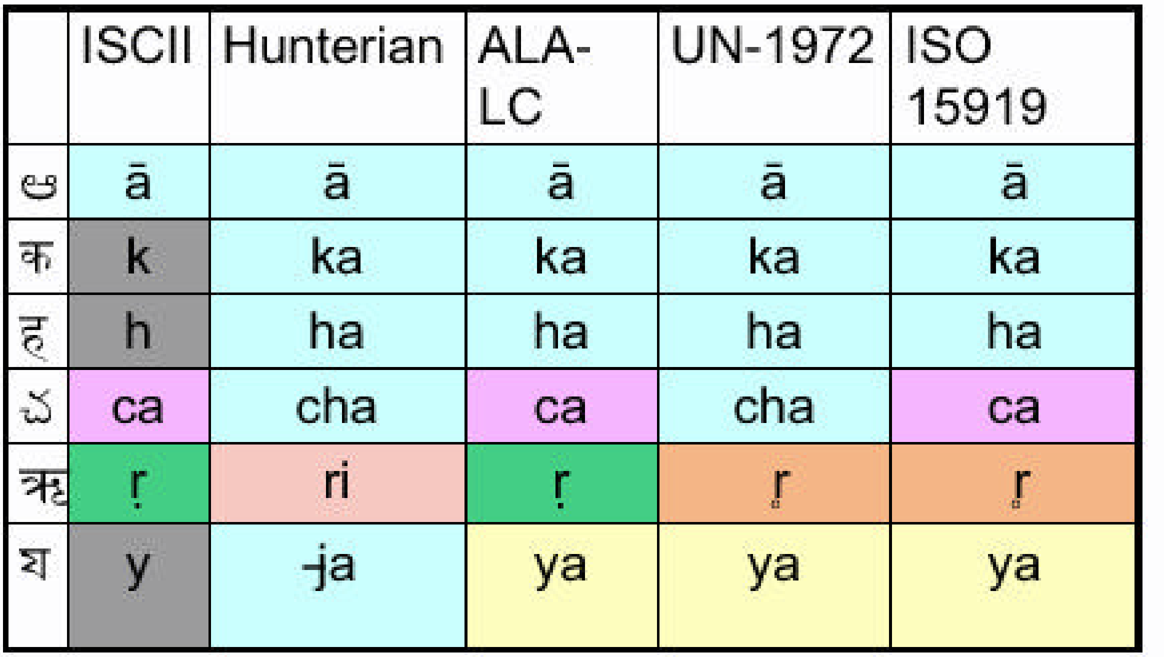
- Implicit vowels at the end of words are not eliminated for simplicity
- Typographic symbols like "&" which represent a word in Latin are also included
- Latin punctuation is preserved except for full stop, which is replaced by DANDA where appropriate (Northern Indic Scripts)

Commonality Among Standards

Salient features in all standards:

- Most Vowels have the same transliterations in all standards expect for some some older vowels
- Most Consonants have the same transliterations in all standards
- Some standards are based on the pronunciation of characters not the semantic value of characters, so there are differences.
- The table above compares Romanizations in different standards.

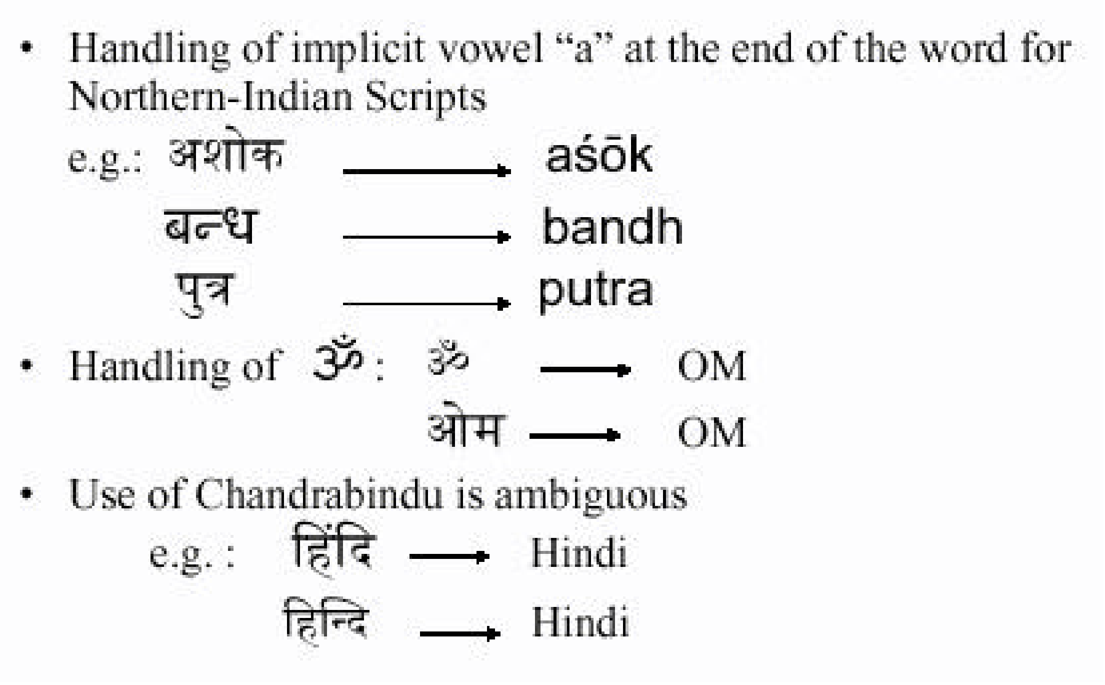
ISCII omits the implicit vowel "a" for Northern Indic Scripts (indicated by gray color)
For southern scripts (indicated by mauve color) the implicit vowel is not omitted.



	ISCII	Hunterian	ALA-LC	UN-1972	ISO 15919
ఆ	ā	ā	ā	ā	ā
क	k	ka	ka	ka	ka
ह	h	ha	ha	ha	ha
చ	ca	cha	ca	cha	ca
ऋ	ṛ	ri	ṛ	r̥	r̥
य	y	ja	ya	ya	ya

Problems in Romanization

As shown the examples above for Northern-Indic scripts like Devanagari, Gujarati the vowel "a" at the end is eliminated (ashok).
The implicit vowel is also eliminated for nasal conjuncts where consonant is preceded by consonants (bandh) and words ending in other conjuncts retain the implicit vowel (putra)
Special rules need to be provided to treat OM when found in isolation as "OM".
The ambiguity in use of Chandrabindu results in loss of reversibilty.



- Handling of implicit vowel "a" at the end of the word for Northern-Indian Scripts

  e.g.: अशोक → aśōk

  बन्ध → bandh

  पुत्र → putra

- Handling of ॐ : ॐ → OM

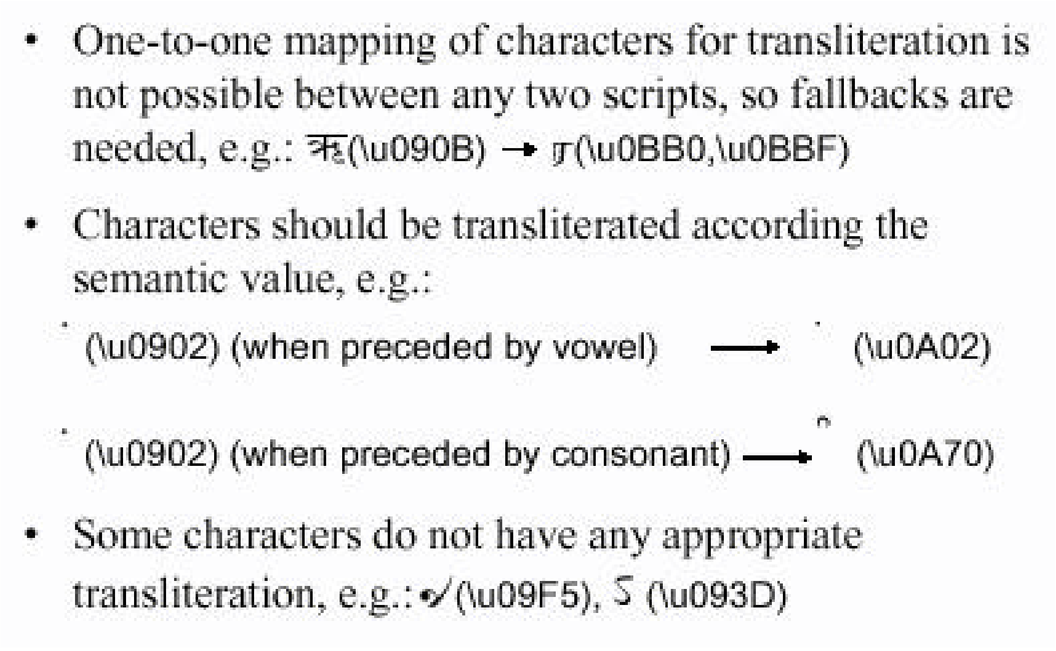
  ओम → OM

- Use of Chandrabindu is ambiguous

  e.g. : हिंदि → Hindi

  हिन्दि → Hindi

Problems in Inter-Indic Transliteration



- One-to-one mapping of characters for transliteration is not possible between any two scripts, so fallbacks are needed, e.g.: ऋ(\u090B) → ரி(\u0BB0,\u0BBF)
- Characters should be transliterated according the semantic value, e.g.:

  ं (\u0902) (when preceded by vowel) ⟶ ਂ (\u0A02)

  ं (\u0902) (when preceded by consonant) ⟶ ੰ (\u0A70)
- Some characters do not have any appropriate transliteration, e.g.: ৵(\u09F5), ऽ (\u093D)

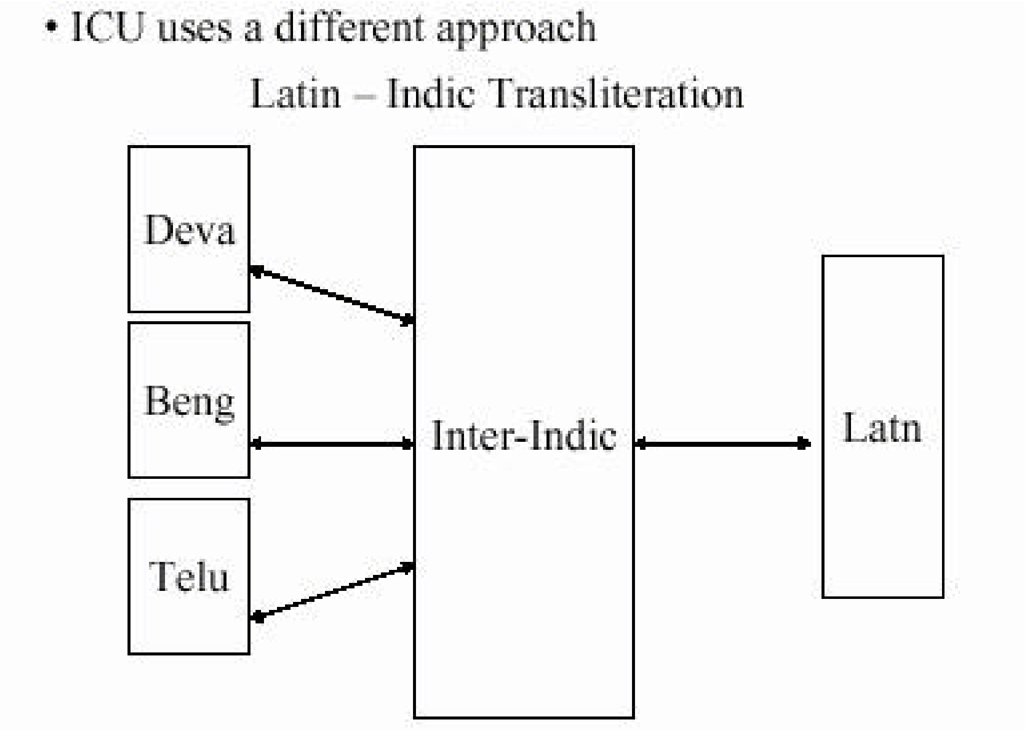
LETTER VOCALIC R is not in Tamil script nor in the Tamil block of Unicode, so when DEVANAGARI LETTER VOCALIC R is transliterated the transliteration is mapped to TAMIL LETTER RA + TAMIL VOWEL SIGN I, which produces the desired pronunciation.
In Gurmukhi, Bindi is used for representing anusvara when combined with the vowels signs and with the independent vowels Tippi is used in all other cases for representing anusvara. Similar rules exist with other scripts which are discussed later.
When characters do not have any appropriate transliteration they should be consumed and not replaced with any other character. This results in partial loss of reversibility, e.g.: BENGALI CURRENCY NUMERATOR, DEVANAGARI SIGN AVAGRAHA.

Implementation approaches

1. Provide transliteration rule sets for all scripts individually
   – Does not take advantage of common underlying structure
   – Increases data since number of rule sets required are 90
2. Treat Devanagari script as superset of Indic Scripts for Inter-Indic transliteration
   – Decreases number of rule sets but many special cases need to be handled
   – May not give correct transliteration for all characters
3. Transliterate Latin to Devanagari and add delta to arrive at the desired Indic script
   – Assumes that placement of characters of Indic Scripts in Unicode is based on the semantic value of the characters

In order to reduce the data required for transliteration, the commonality of structure needs to be considered. Individual rule sets can transliterate accurately but data required is large which leads to problems like size and maintenance.

Implementation in ICU



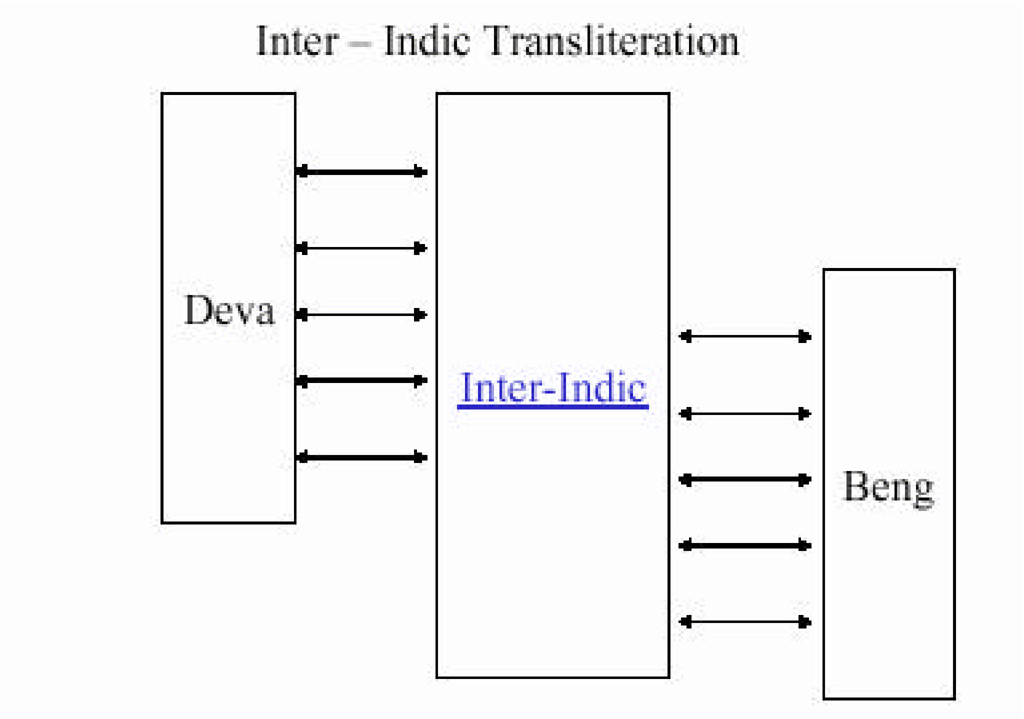
- ICU uses a different approach

Latin – Indic Transliteration

ICU uses a portion of Private Use Area (PUA) region for transliteration.
Reasons for choosing the PUA approach:
•There are 9 Indic scripts in Unicode, to transliterate them to and from Latin individually, we would require 18 different rule sets.
•If the Inter-Indic transliterations are also considered, the number of rule sets for transliteration will be 72.
•All Indic scripts share an underlying structure owning to their origin in the Brahmi script. This commonality can be leveraged to reduce the number of rule sets and hence data.
•None of the Indic scripts is a true superset of all other scripts as shown in the next chart.
So we used PUA range of \uE000-\uE007F to represent a true superset of the all Indic scripts.
We now have (9 Indic scripts + Latin) x 2 = 20 rule sets instead of 90.

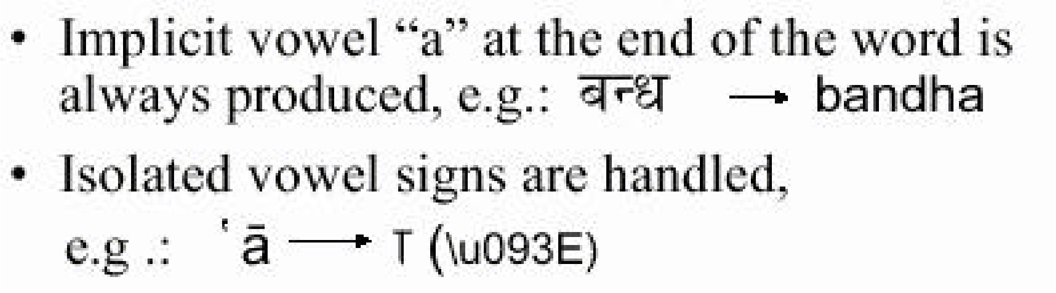
Implementation in ICU (Contd.)

For Inter-Indic transliteration ICU uses a script-neutral pivot range in Unicode.
The above figure shows the relationship between Devanagari and Bengali with respect to Inter-Indic block. There are some characters in The link provided above points to the comparison chart that brings out the complex relationship between characters in Indic scripts.
Color code for the chart:
PINK: Inter-Indic PUA codepoint
GREEN: Inter-Indic – Latin transliteration
WHITE: The codepoint is roundtripped
BLUE: The codepoint is transliterated to the character shown as a fallback
YELLOW: The codepoint is eliminated during transliteration to/from the particular script
During the presentation the chart is explained and important features are highlighted.



Romanization of Indic Scripts

• ICU conforms to ISO 15919 standard for the most part except for
  – Transliteration of typographical symbols
  – Extra accents are used for distinguishing some characters



- Implicit vowel “a” at the end of the word is always produced, e.g.: बन्ध → bandha
- Isolated vowel signs are handled, e.g .: 'ā → ा (\u093E)

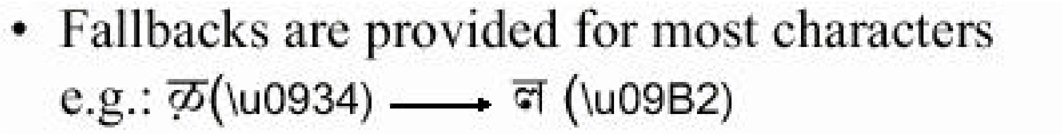
Indic Transliteration is implemented using pseudo super set of Indic Scripts mapped to the Private Use Area block of Unicode.
Romanization of Indic scripts conforms to the ISO 15919 standard expect for handling of typographical symbols.
The ISO 15919 standard allows for ignoring implicit vowel “a”. To implement this feature the transliterator requires

i) A dictionary based lookup mechanism to find which words are nouns
ii) Conform to rules for elimination of implicit vowel from ISCII standard
as stated previously

Reversibility of transliterations is lost in the process.
ICU ignores vowel elimination and produces it always.

Other Features

• All canonically equivalent text is handled correctly
• Rule Based: easy to customize differently



- Fallbacks are provided for most characters
  e.g.: ऴ(\u0934) ⟶ ল (\u09B2)

• Characters are eliminated if no appropriate transliteration or fallback is available

All characters either in pre-composed or de-composed form, if they are canonically equivalent will be transliterated correctly.
All transliterators are rule based. The rule syntax is presented and important features are highlighted
Fallbacks:
- Approximate pronunciation can be figured out.
- Results in loss of reversibility

Elimination of characters that do not transliterate correctly to a target script is better than having text in target script sprinkled with characters from source script.

Demo

http://oss.software.ibm.com/cgi-bin/icu/tr/
During the presentation the transliteration engine is demonstrated.

Conclusion

• Romanization of Indic scripts can be achieved by using a superset
• Special cases and special rules for transliteration of Inter-Indic scripts should be handled
• Other approaches presented, while feasible have drawbacks
The implementation in ICU has demonstrated that it is possible to use PUA region as pivot and superset for Indic scripts and perform transliteration for all combinations of scripts with a comparatively small number of rule sets
Special cases and all characters in source script that have reasonable transliterations are transliterated.
Number of rule sets are reduced and hence the data required for performing transliterations.

References and Resources

• How to use ISO 15919:
http://homepage.ntlworld.com/stone-catend/translit.htm
• Transliteration of non-Roman Alphabets and Scripts:
http://homepage.mac.com/sirbinks/
• Indian Scripts and Unicode:
http://members.tripod.com/~jhellingman/IndianScriptsUnicode.html
• International Components of Unicode (ICU):
http://oss.software.ibm.com/icu/
• Unicode Consortium: http://www.unicode.org
• IBM developerWorks:
http://www.ibm.com/developerworks/unicode