Introduction to Indian language computing

MMP – e-district

Most Computer System, Solutions and devices even today are basically designed and developed for English.

- We are trying to change this mindset by covering MMPs at design and RFP stage

Localisation of applications, data, reports, code, services, devices... for Indian Languages

All systems can be broken down into

Input

Storage / processing

Output

Applications for Indian Languages should have support throughout the lifecycle of the system – rather than being an after thought.

Input – INSCRIPT std, Phonetic / transliteration, Typewriter, limited keys on devices

Storage / processing – UNICODE defacto even though it is more expensive

Output – Fonts, screen resolutions, printer and display device

One script: many languages

Devanagari – Hindi, Marathi, Konkani, Rajasthani, Sindhi, Nepali, Dogri, Santhali, etc. Thus the data in Devanagari (code page) can support all languages using that particular script. However tools like synonym Dictionaries, spellcheckers, and search engine crawlers and indexers, etc. are language dependent and require language information along with the data.

Though the contents would reveal the language used, it would be ideal if a special attribute code to indicate the language is inserted.

One language: many scripts

Konkani is written in Roman, Devanagari, Malayalam and Kannada. Sindhi is written in Gurmukhi (Punjabi), Arabi (Perso-Arabic), Devanagari, Gujarati and also Roman. Sindhi has adopted the Perso-Arabic script for representing their language. In case of Konkani, Devanagari is used as official script.

Language	ISO	Official Language	Family	Script
Assamese	asm	Assam	Indo-Aryan	Assamese
Bengali	ben	Tripura and West Bengal	Indo-Aryan	Bangla
Manipuri	mni	Meitei	Tibeto-Burman	Bangla Meitei-Meyek
Boro	brx	Assam	Tibeto-Burman	Devanāgarī (modified)
Dogri	dgo	Jammu and Kashmir	Indo-Aryan	Devanāgarī (modified)
Hindi	hin	Andaman and Nicobar Islands, Bihar, Chandigarh, Chhattisgarh, Delhi, Haryana, Himachal Pradesh, Jharkhand, Madhya Pradesh, Rajasthan, Uttar Pradesh and Uttaranchal	Indo-Aryan	Devanāgarī
Konkani	kok	Goa	Indo-Aryan	Devanāgarī Roman (Latin)
Maithili	mai	Bihar	Indo-Aryan	Devanāgarī
Marathi	mar	Maharashtra	Indo-Aryan	Devanāgarī
Nepali	nep	Sikkim	Indo-Aryan	Devanāgarī

Sanskrit	san	Pan-Indian	Indo-Aryan	Devanāgarī
Gujarati	guj	Dadra and Nagar Haveli, Daman and Diu, and Gujarat	Indo-Aryan	Gujarati
Punjabi	pan	Punjab	Indo-Aryan	Gurmukhi
Kannada	kan	Karnataka	Dravidian	Kannada
Malayalam	mal	Kerala and Lakshadweep	Dravidian	Malayalam
Santali	sat	Jharkhand	Munda	Ol Ciki
Oriya	ori	Orissa	Indo-Aryan	Oriya
Kashmiri	kas		Indo-Aryan	Perso-Arabic Devanāgarī
Sindhi	snd	Pan-Indian	Indo-Aryan	Perso-Arabic Devanāgarī Gujarati Roman (Latin)
Urdu	urd	Jammu and Kashmir	Indo-Aryan	Perso-Arabic
Tamil	tam	Tamil Nadu and Pondicherry	Dravidian	Tamil
Telugu	tel	Andhra Pradesh	Dravidian	Telugu

Complexities in Indian languages and Computers:

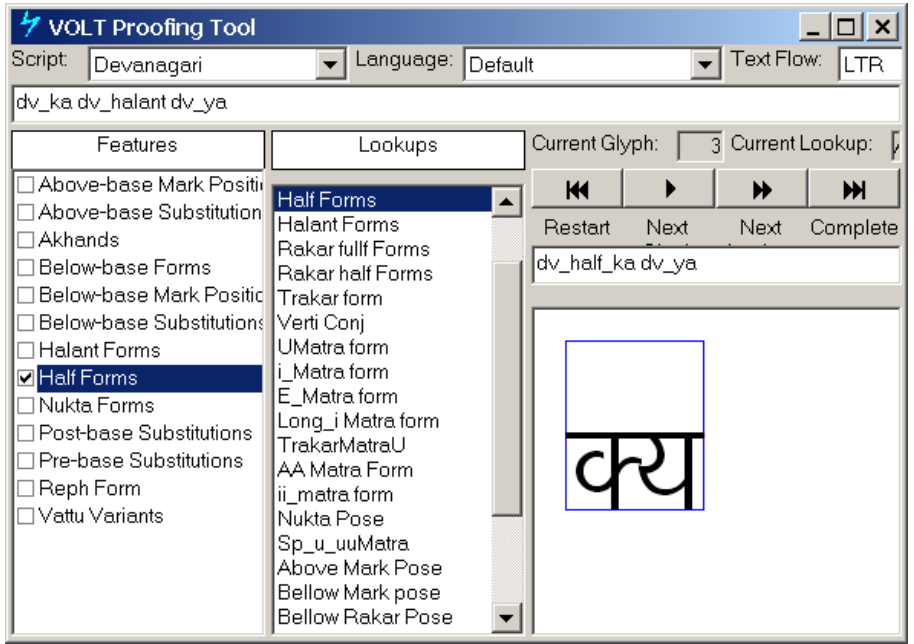
In English script, there is one to one correspondence between what you type, what you store & what you display. Also it is a linearly mapped according to the character set. It does not grow vertically like Devanagari or Arabic script. Which necessarily means that if 'A' is inputted from the keyboard then internally the code of 'A' which is 65 in ASCII is stored, while displaying, the Glyph / shape of the letter 'A' placed at the position 65 in the font is used for rendering.

Life is not as simple as English while using Indian languages. Due to the highly complex structure of the Indian languages, there is no one to one correspondence between what you store (character code) & what you display (font code). The relationship between characters and glyphs can be one-to-one or one-to-many or many-to-one. Formation of multiple shapes because of nature of the scripts most of the scripts have a peculiarity that most of its characters can have multiple visual representations depending upon its position in the word. This can be seen in case of Arabic script wherein the character shape has four different forms depending upon whether the character is in beginning, middle, end or standalone.

GIST Graphics and Intelligence based Script Technology

Complexities of Indian languages

- Research and development 22+ Indian languages including the right to left scripts of Urdu(Naskh and Nastaleeq), Sindhi and Kashmiri
- GIST has been involved in development of highly calligraphic True Type, Open Type and Bitmap Fonts for various media such as Desktop – for screen as well as printing, Web media, Broadcast / Television media, Embedded and Mobile Computing
- Compared to Roman scripts Indian language fonts are very complex. Most of them have multi-tier system



Pre-Unicode Era

ा = ◌ + ा

त = ◌ + ा

व = ◌ + ा

श = श + ा

शीं = श + ा + ीं

Advantages of Font Encoding

- Easy to use
- Easy for the people to understand
- Easy to provide solution on existing technology

Disadvantages of Font Encoding

- Proprietary to the developer.
- Standards like ISO font encoding standard are seldo used, which leads to problems in printing and displaying.
- Limited searching capabilities.
- Sorting is not possible.
- May not create unique spelling (p = p or P + a).
- Glyph based approach and not character based.

Converter are required to convert the data to any other format which may result in loss of data

UNICODE

- STORAGE STANDARD
- WHAT ASCII IS FOR ENGLISH, UNICODE IS FOR OTHER LANGUAGES OF THE WORLD
- ENABLES SEAMLESS EXCHANGE OF DATA – DESKTOPS, PRINTERS, DATABASES, BROWSERS, DEVICES

UNICODE

- Unicode consortium defines Unicode as :

- *“Unicode is the universal character encoding, maintained by the Unicode consortium. This encoding standard provides the basis for processing, storage and interchange of text data in any language in all modern software and information technology protocols.”*

- It is the superset of all the languages in the world which also includes punctuation, special characters (shapes), currency symbols, mathematical symbols etc. Using Unicode, more than 65000 different characters can be represented. Unicode comprises of many code pages.
- The Unicode code charts can be referred at: http://www.unicode.org/charts/.

UNICODE

- Various editors / applications / development environments / databases / browsers need to understand how to read in the given Unicode data and interpret the same. Various encoding schemes to represent Unicode are UTF-8, UTF-16, UTF-32 with a combination of endian-ness.

- There are normalization rules which are required to be followed for data compatibility between various applications / underlying environment. Non adherence to some of these may lead to wrong interpretation of data and will also pose problems in searches as well.

UNICODE

- UNICODE is a 16 bit character based encoding standard.
  - A mapping of characters to numbers
  - Syntax rules for display of complex scripts
  - Not a font or glyph encoding!
  - Not a sort algorithm!

- Includes **all** characters in common use in modern scripts (and others)

Character semantics

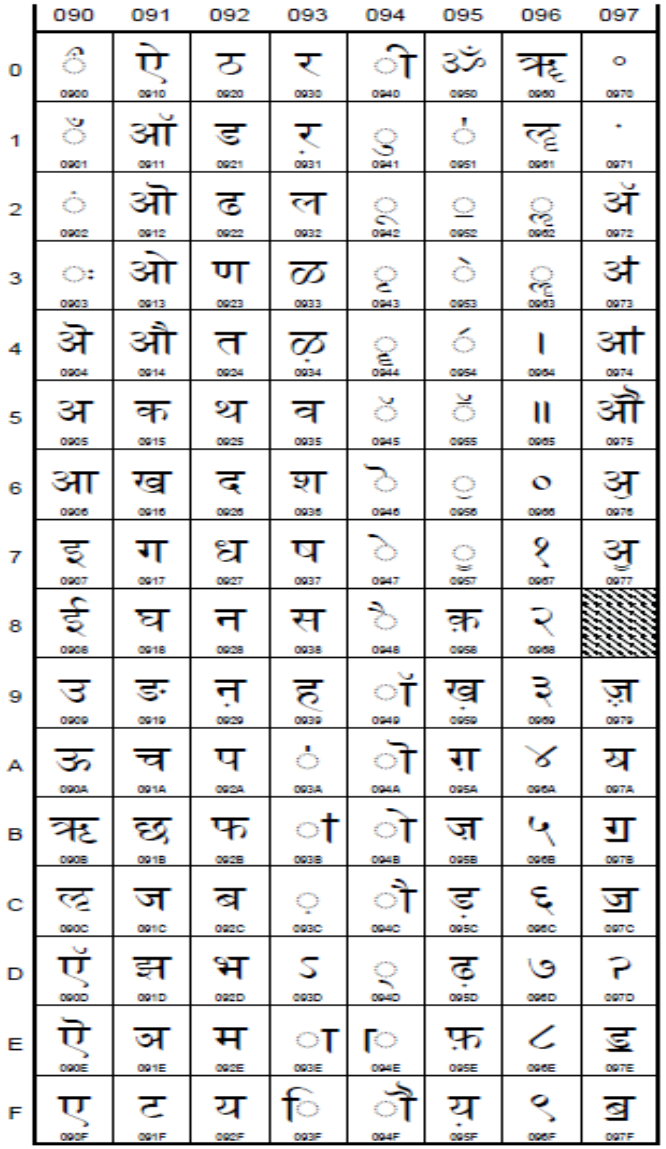
- The Unicode standard includes an extensive database that specifies a large number of *character properties,* including:
  - Name
  - Type (e.g., letter, digit, punctuation mark)
  - Decomposition
  - Case and case mappings (for cased letters)
  - Numeric value (for digits and numerals)
  - Combining class (for combining characters)
  - Directionality
  - Line-breaking behavior
  - Cursive joining behavior
  - For Chinese characters, mappings to various other standards and many other properties

Advantages of UNICODE

- Character based encoding.
- Unicode values are governed by characters (vowels and consonants).
- Can be ported on any platform and any OS.
- Can be ported on hand held and mobile devices
- Different scripts have different code page.
- All Indian languages are supported along with all other languages.
- Allows multiple languages in the same data.

UNICODE Devanagari Code Page

0900 Devanagari 097F



	090	091	092	093	094	095	096	097
0	ऀ 0900	ऐ 0910	ठ 0920	र 0930	ी 0940	ॐ 0950	ॠ 0960	० 0970
1	ँ 0901	ऑ 0911	ड 0921	ऱ 0931	ु 0941	॑ 0951	ॡ 0961	ॱ 0971
2	ं 0902	ओ 0912	ढ 0922	ल 0932	ू 0942	॒ 0952	ॢ 0962	ॲ 0972
3	ः 0903	ओ 0913	ण 0923	ळ 0933	ृ 0943	॓ 0953	ॣ 0963	ऄ 0973
4	ऄ 0904	औ 0914	त 0924	ऴ 0934	ॄ 0944	॔ 0954	। 0964	आ 0974
5	अ 0905	क 0915	थ 0925	व 0935	ॅ 0945	ॕ 0955	॥ 0965	औ 0975
6	आ 0906	ख 0916	द 0926	श 0936	ॆ 0946	ॖ 0956	० 0966	ॵ 0976
7	इ 0907	ग 0917	ध 0927	ष 0937	े 0947	ॗ 0957	१ 0967	ॶ 0977
8	ई 0908	घ 0918	न 0928	स 0938	ै 0948	क़ 0958	२ 0968	
9	उ 0909	ङ 0919	ऩ 0929	ह 0939	ॉ 0949	ख़ 0959	३ 0969	ॹ 0979
A	ऊ 090A	च 091A	प 092A	़ 093A	ॊ 094A	ग़ 095A	४ 096A	ॺ 097A
B	ऋ 090B	छ 091B	फ 092B	ा 093B	ो 094B	ज़ 095B	५ 096B	ॻ 097B
C	ऌ 090C	ज 091C	ब 092C	ि 093C	ौ 094C	ड़ 095C	६ 096C	ॼ 097C
D	ऍ 090D	झ 091D	भ 092D	ी 093D	् 094D	ढ़ 095D	७ 096D	ॽ 097D
E	ऎ 090E	ञ 091E	म 092E	ु 093E	ॎ 094E	फ़ 095E	८ 096E	ॾ 097E
F	ए 090F	ट 091F	य 092F	ू 093F	ॏ 094F	य़ 095F	९ 096F	ॿ 097F

Availability

- UNICODE is not vendor specific
- Backward compatible
- Major database, OS, browser players support some form UNICODE encoding
- Data Migration services will be provided free for e-governance developers
  - .doc, .xls, can be converted to UNICODE

Enhanced INSCRIPT (2.0)

- INSCRIPT is part of BIS standard – ISCII
- Enhanced INSCRIPT allows user to type latest UNICODE characters such as Rupee symbol.
- Unlike the phonetic or transliteration mechanism, it does not expect the user to know English to type Indian language and so caters to rural audiences as well.
- Fast typing is possible as consonants are typed by one hand while vowels are typed by left hand

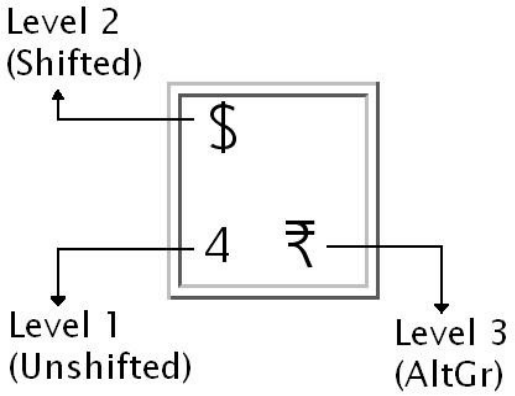
Enhanced INSCRIPT Standardization for Latest UNICODE

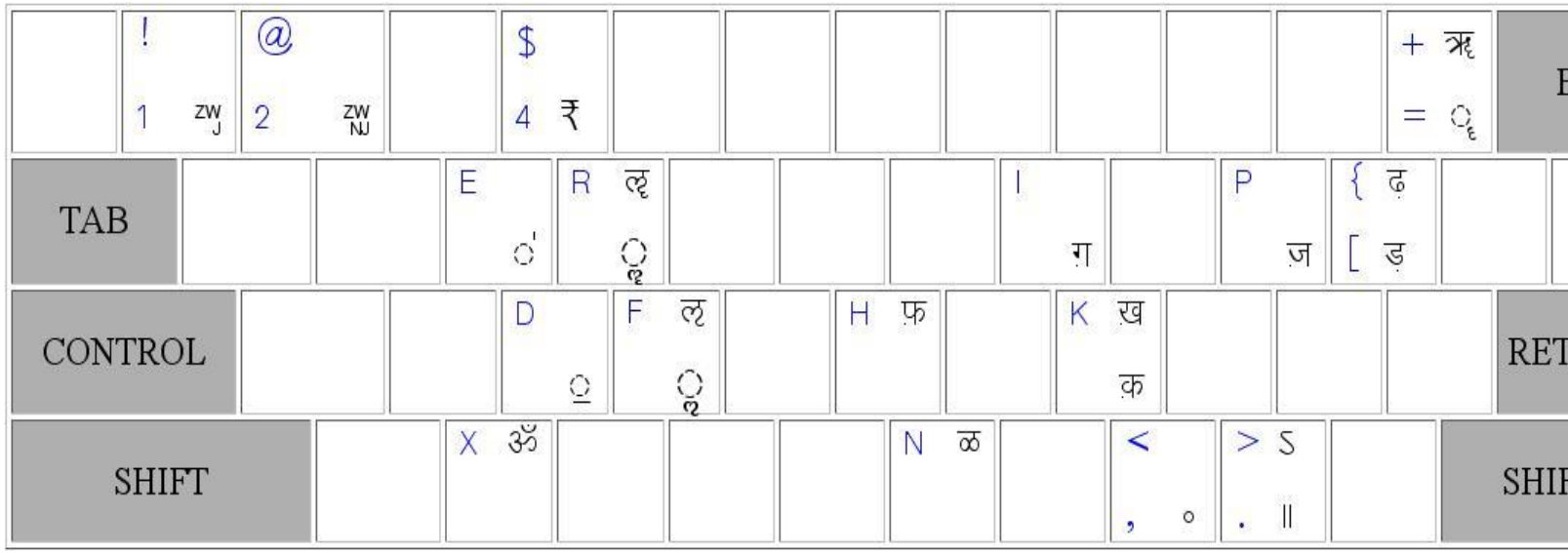
- **Study and Research for Keyboards of various languages Normal layer and Extended layer**

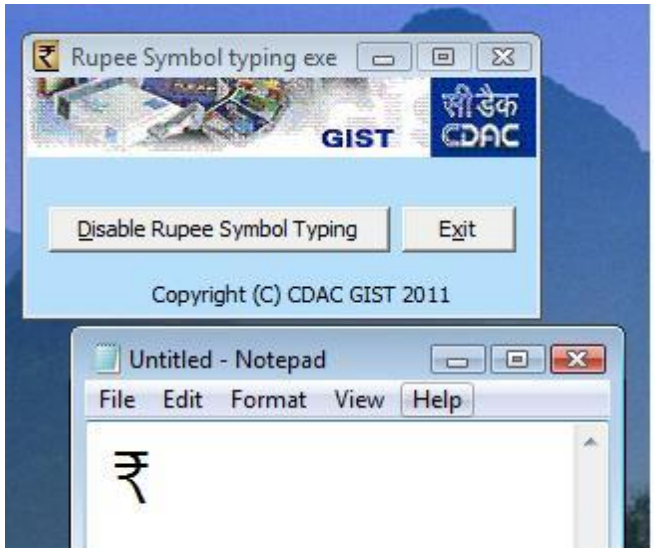
- **Along with teams from – Microsoft, Redhat and IBM**

The Enhanced INSCRIPT keyboard layout provides three layers and this to accommodate all the extra characters and yet make the keyboard as ergonomic and efficient as possible

Standardization of Rupee Symbol Inputting







Made available for free download on
http://tdil-dc.in

Syllable (Akshar) Based Cursor Movement, addition and deletion

- Cursor movement and deletion of characters should be based on syllables. A syllable is a unit of organized sequence of code points. The structure of the written syllable (akshar) is defined as per ISCII (IS 13194 : 1991).

- Lets take an example string किताब

Basic Inputting

Lets start with a basic word

“सीडैक”

Basic Inputting

Basic Characters

- Steps to be followed to input सीडैक:

1. Logically note the sequence of characters in word सीडैक as you would pronounce.
2. You may note that, we pronounce it as

   “sa-i-da-ae-ka”

3. Thus, the inputting sequence becomes

   “स-ी-ड-ै-क”

Basic Inputting

Halanta

- To form conjuncts, a special character halanta (“्”) is used.
- Lets try one: e.g. प्रस्थान
  - Logical pronunciation sequence

    “**PRA**-**STHA**-na”
  - Note that there are no special PRA and STHA on keyboard, since they are conjuncts
  - Such conjuncts are created with the help of halanta (“्”)

Basic Inputting

Halanta

- Case: प्रस्थान
  - Logical pronunciation sequence

    “**PRA**-**STHA**-na”
  - Lets create PRA and STHA

    PRA – Pa+Ra

    KHYA – Sa+Tha
  - Since, we join two characters with halanta (“्”)

    PRA – Pa+Ra – Pa+Halanta+Ra (प+्+र)

    STHA – Sa+Tha – Sa+Halanta+Tha (स+्+थ)
  - Thus, प्रस्थान gets inputted as,

    “**PRA**-**STHA**-na”

    (प+्+र + स+्+थ +ा+न)

Basic Inputting

Reph and **Rakar** Cases

- Reph and Rakar are special combinations with र character
- Case: क्र

  - Logical pronunciation sequence

    “KRA”

  - It is a combination of क and र, but the **sound of Ka comes before Ra**

    KRA –Ka+Ra

  - Since, we join two characters with halanta (“्”)

    KRA – Ka+Ra – Ka+Halanta+Ra (क+्+र) – क्र

Such a conjunct of Ra in which Ra is latter part of conjunct, is called as “**Rakar**”

Basic Inputting

Reph and Rakar Cases

- Reph and Rakar are special combinations with र character
- Case: र्क

  - Logical pronunciation sequence

    “RKA”

  - It is a combination of र and क, but the **sound of Ka comes after Ra**

    RKA –Ra+Ka

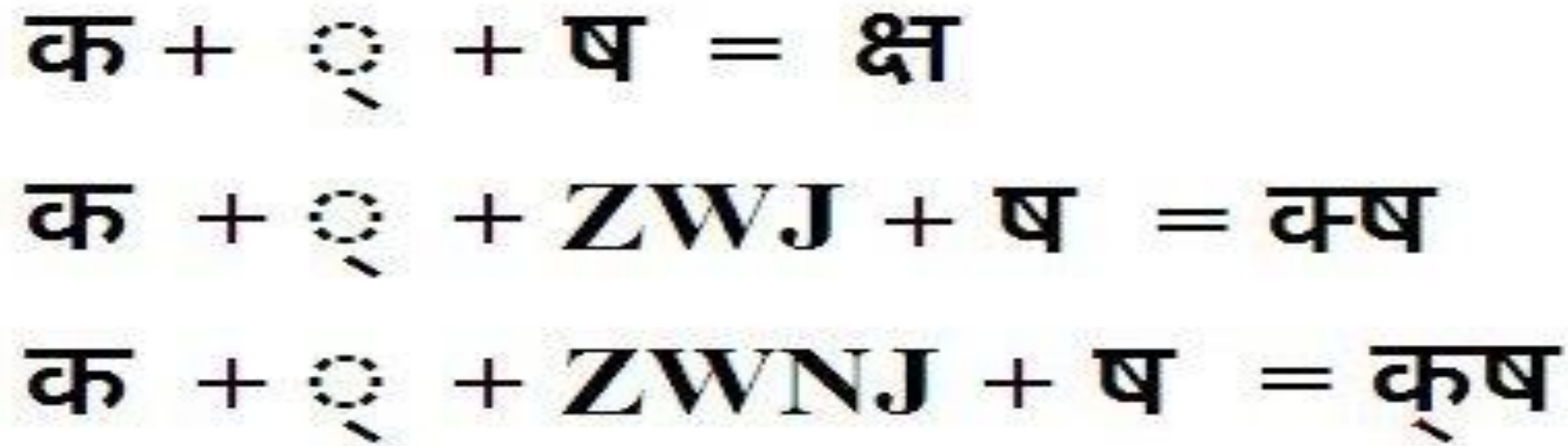
  - Since, we join two characters with halanta (“्”)

    RKA – Ra+Ka – Ra+Halanta+Ka (र+्+क) – र्क

Such a conjunct of Ra in which Ra is former part of conjunct, is called as “**Reph**”

ZWJ and ZWNJ

- Two special characters in Unicode
- ZWJ - 200D, ZWNJ - 200C



Availability

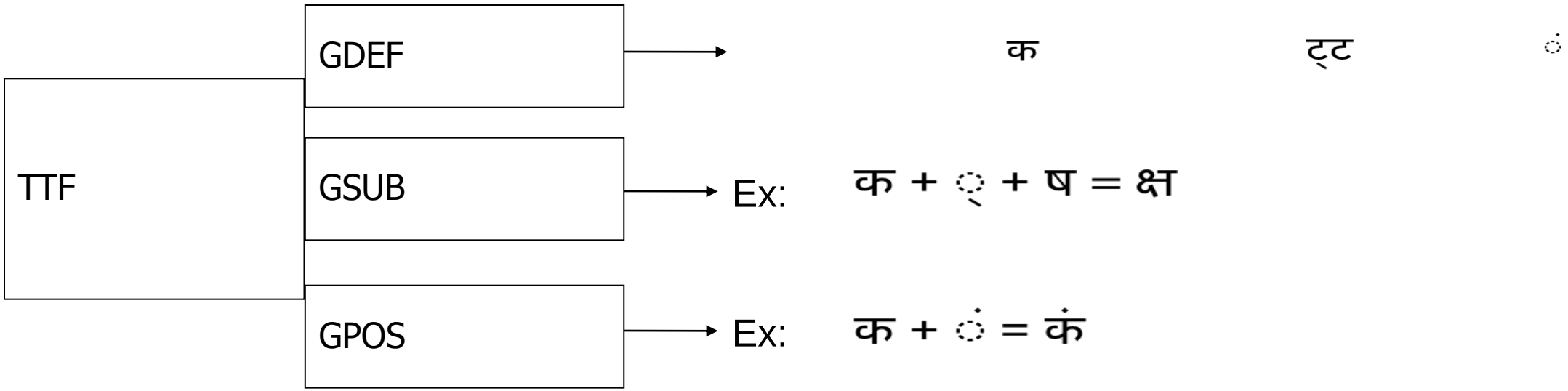
- UNICODE typing Tool is available for free download from http://ildc.in
- It has all 22 languages and supports enhanced INSCRIPT layout including the Rs. Symbol
- The keyboard sticker layouts are also available for download from http://ildc.in
- Onscreen Javascript for websites will be made available free of cost to all e-governance developers under the project

OpenType Fonts

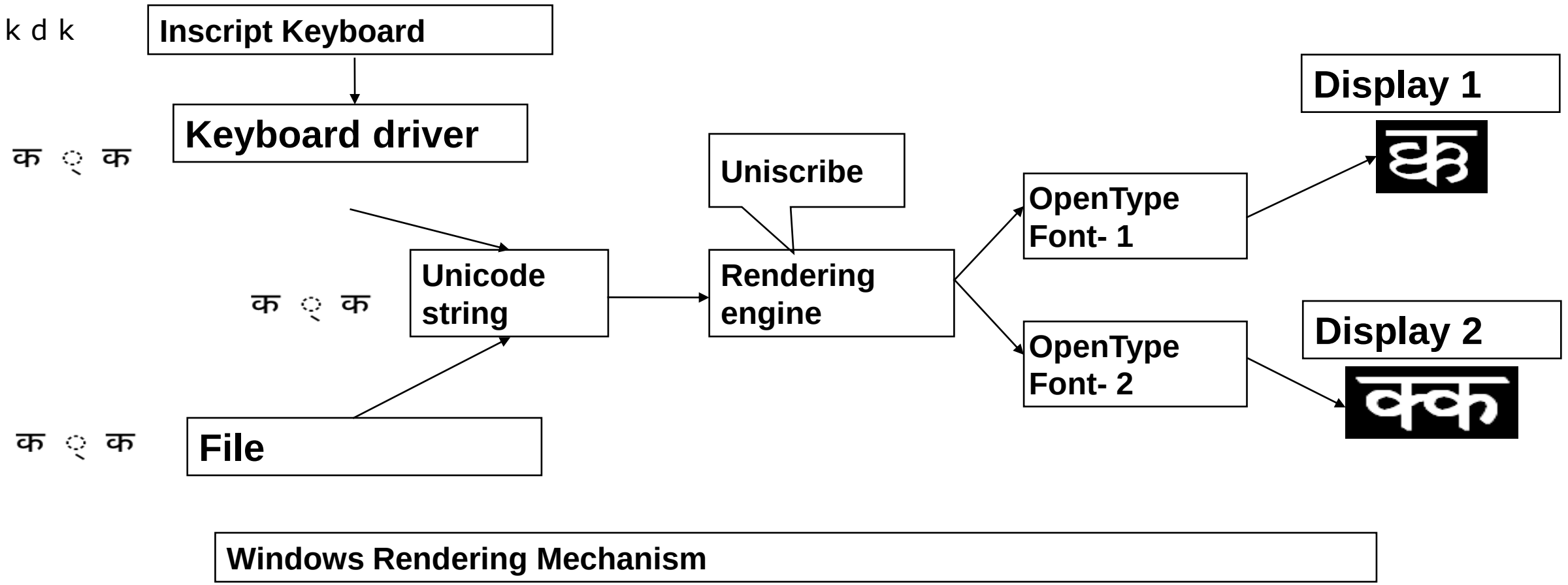
- Joint effort by Adobe and Microsoft
- 16-bit Unicode compliant, more glyphs possible
- Glyph substitution & positioning logic built into the font
- Storage-to-display conversion is done by the rendering engine
- Data is not stored in glyph codes rather in Unicode
- No issue of data portability
- No need to have a font glyph standard

Open Type Font

Ex: (Simple , Ligature, Mark)



Unicode and OpenType Fonts



Searching in Indian languages

Several words have **multiple correct spellings** and
Alternate representation forms

eg: the word Hindi may be written
with a bindi on top of the first syllable or with a half na.

हिंदी हिन्दी

What should happen in case of using database querries

So also with the representations of the word vitthal

विट्ठल विठ्ठल

Normalization in Unicode

- The Unicode data requires normalization. There are many cases where a character can be entered in more than one ways using the Unicode code chart. If application or database does not normalize searching becomes difficult

रिज़र्व

ज + ◌़ = 091C+093C

ज़ = 095B

रिज़र्व = र + ि + ज + ़ + र + ् + व

Also = र + ि + ज़ + र + ् + व

Terminology in user interface

Localisation of strings

- Translation v/s Transliteration
- Technical Term v/s common man's Term
- Physical-size of localised equivalent strings

- 3 out of 22 languages are right to left oriented
- Location / Layout
  - Positioning of back-next buttons, scroll bar positions for applications supporting right to left scripts.
  - coexisting along with English (Bi-directional support)

Localisation of strings

- Context and Domain specific meanings
  - Example
  - the word ‘Bank’ (Financial Entity, River bank, to trust on someone/thing, etc.)
  - the word ‘Fire’ (may very in meaning depending on context) – If it is as a verb (such as fire an event) then it may suggest some action to be undertaken, If noun then the meaning changes completely
  - Multi-Domain expertise as well as context may be required apart from linguistic know-how

Localisation of strings

- Technical terminology
  - Differentiating between similar meaning such as cancel, abort, terminate
  - Translation v/s Transliteration (IPR and registered copyrights and trademarks)

  - What should be Localised string for :
    - Windows
    - Mouse
    - FireFox
    - Internet Explorer
    - Double click
    - Dock Windows

Getting consensus is difficult

FUEL

- FUEL is an open source initiative to standardize terms for open source software programs. The GIST Group of CDAC is actively participating in the same and has initiated FUEL for “Web”, “Standalone Applications” and “Mobile” platforms. It aims at resolving the problem of term inconsistency and lack of standardization in Computer software translation, across various platforms. It also works to provide a standard and consistent terminology for a language. Following Indian language support has been added in this initiative. Following languages are being covered under the FUEL. Remaining languages work is in progresss.
- Assamese, Bengali (India)
- Gujarati, Hindi
- Maithili , Malayalam
- Marathi , Punjabi
- Oriya , Tamil
- Telugu , Urdu
- Kannada

Common Locale Data Repository (CLDR)

- The CLDR provides key building blocks for software to support the world's languages. The data in the repository is used by companies for their software internationalization and localization: adapting software to the conventions of different languages for such tasks as formatting of dates, times, time zones, numbers, and currency values; sorting text; choosing languages or countries by name; and many others. C.L.D.R.’s provide useful information as to the locale and are therefore crucial from the perspective of localization. Mobile based CLDRs should be made and used to enhance the localisation across different cultures and locales. CLDR mostly comprises of
- Calendars
- Numeric formats,
- Date and Time formats
- Currencies

Sample Extract of Dogri CLDR

- <monthWidth type="wide">
- <month type="1">जनवरी </month>
- <month type="2"> फरवरी </month>
- <month type="3">मार्च</month>
- <month type="4">एप्रैल </month>
- <month type="5">मेई </month>
- <month type="6">जून </month>
- <month type="7">जूलै </month>
- <month type="8">अगस्त </month>
- <month type="9">सितंबर </month>
- <month type="10">अक्तूबर </month>
- <month type="11">नवंबर </month>
- <month type="12">दिसंबर </month>
- For further details please refer: http://cldr.unicode.org/

- LPMS movie

- Thank you