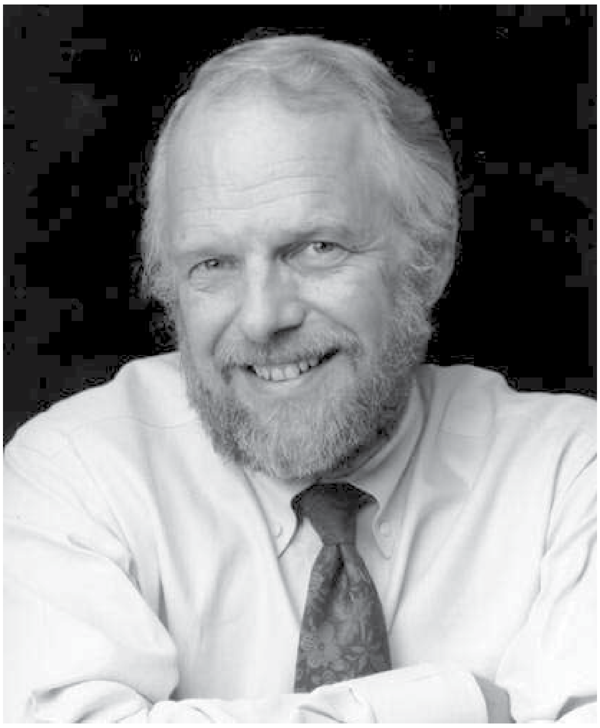
MIGRATING TO UNICODE FROM LEGACY SYSTEMS

A PRACTICAL GUIDE FOR DTP TYPESETTERS AND PUBLISHERS

by
Nasim Ali

> “The biggest enemy of a better system is an existing system that just works.”
> —*Dennis Ritchie*

This page intentionally left blank



John E. Warnock

Charles M. Geschke

Dedication

This booklet is dedicated to Dr. John E. Warnock and Charles M. Geschke, former engineers at Xerox PARC who invented PostScript and founded Adobe Systems to sell it. Steve Jobs licensed PostScript technology for Apple's LaserWriter printers, which combined with Apple's Macintosh and Aldus' Pagemaker, completely changed the Printing industry. It also saved Mac from impending near-certain extinction and cemented its place as the platform of choice for creative designers for decades to come.

Contents

Contact info:

You can contact me at nasimali2008[at]gmail.com

Some Conventions used in this document:

- ◊ Legacy systems: Any old system for displaying and printing Indic text (including Odia) that doesn't comply with the Unicode conventions. Examples are Akruti, ShreeLipi (including 7.0 Ex fonts), iLeap etc.
- ◊ SP2/3: Service Pack 2/3 for Windows XP
- ◊ ShreeLipi: Modular System's ShreeLipi v4/5/6. When I refer to older versions or the newer 7 version, I mention that specificlly.

A companion video for this guide is available on YouTube.

Updated versions of this booklet will be hosted on GitHub.

document version 1.01 (30122015-2046)

1. Unicode

1.1 A brief history

Computer technology was developed in the Western countries, more specifically USA. So early computers mostly used English for all practical purposes of user interaction. Even countries like Japan had trouble with bringing their own languages to computer systems. The important issue was the complexity of non-Latin scripts especially Asian ones.

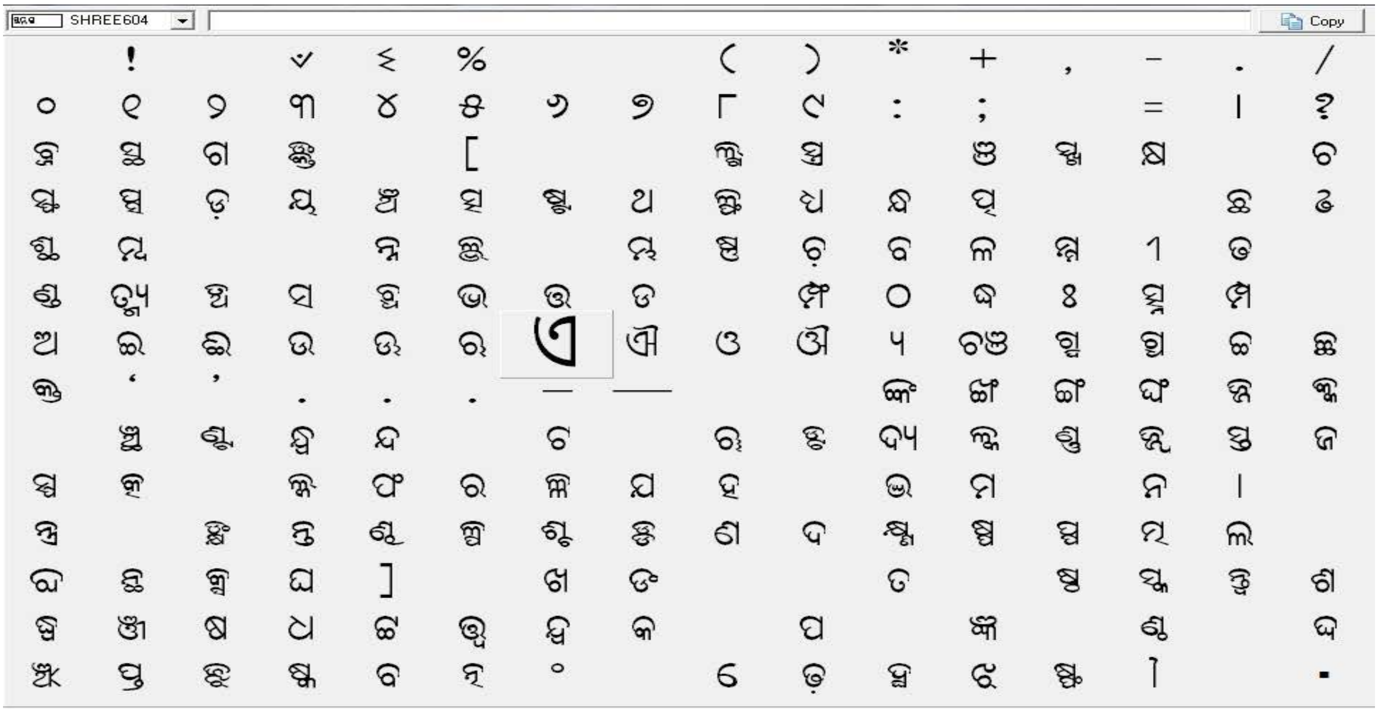
In the late 70s Desk Top Publishing was conceived as an alternative to large scale printing machines. It came to life in Mid 80s with Aldus' Pagemaker (DTP term was itself coined by Aldus CEO Paul Brainerd). Apple's computers and printers revolutionized printing industry. Initially what was a means of printing for personal and small business purpose became big. From Gutenberg's movable types to Linotype machines to digital printing: the typesetting industry has undergone several paradigm changes, but this was the biggest yet.

The computers of that era didn't support any Indian languages. So to input Indic text for DTP or otherwise, clever compromises were made. Fonts were made with characters mapped to ASCII character set. So the computer stored data that were actually Latin (extended) characters but with these fonts selected they appeared as Hindi/Oriya etc. Original ASCII character set was very constrained thus these fonts couldn't hold the large number of characters required for representing Indic languages. This necessitated sacrificing some conjuncts or using multiple font files. This is a good example of what we call 'jugaad' in desi term: using tools to accomplish things that were not designed for that purpose by slight modifications. These worked better than traditional printing. However since the actual underlying text was garbled unreadable ASCII, data exchange was impossible unless sender and recipient had the same font file (see Mojibake).

The first Hindi font was reportedly created by Dataflow systems in 1986. What followed were a number of proprietary DTP products from different vendors, each storing data in their own encodings incompatible with one another. There were efforts by Government of India to standardise one encoding so data exchange would be possible: ISCII was born. However practically no one adopted ISCII, even software from government subsidiaries like GIST used different encodings.

Indic scripts were not the only sufferers of such segmentation. To solve this problem and establish a universal standard, Unicode was born in 1991. It defines a standard encoding for representing all the scripts of the World. Yes, ALL!

Unicode is an open, universal standard supported almost everywhere. Windows, Linux, Mac, Unix, MS Word, Wordpad, Java: almost all modern pieces of software support Unicode. This makes information exchange very easy. Also Unicode will always remain backwards compatible. Currently in version 8, any Unicode text written in version 1 will still appear the same. Compare this to the horror stories of DTP typesetters migrating from ShreeLipi 3.0 to 4.0 or 6.0 to 7.0. Ever experienced the frustration of ଊ breaking in Akruti documents opening in XP. That's where an open standard helps.



[The character map of an old ShreeLipi 8 bit Type-1 PostScript font with Odia characters mapped to ASCII codepoints. The whole font contains only these many characters (plus one more line of characters not seen here). Modern Unicode Odia fonts usually contain twice as many letters and may also contain matching Latin letters.]

1.2 Unicode support vs. CTL support

Before delving further let us understand two technical terms. What's an encoding? It's an agreed or standardized way for representing data. For example in Morse code one dot and one dash represents 'a' while 'o' is represented by three dashes. Similarly think of sign language for deaf and dumb people. Even our written alphabet itself can be considered as an encoding. In computer world the two commonest encodings are ASCII (1963) and Unicode (1991). The way one encoding stores data may not be compatible with others. For example text written in Akruti will not be readable in iLeap or ShreeLipi. One glaring exception is Unicode which is seamlessly backwards compatible with the much older ASCII (you really should ask Modular if Unicode consortium can support an encoding made in 1963 by another entity, why can't Modular support its own encoding from 5 years back and make their upgrades backwards compatible.)

A codepoint is the code assigned to a character or symbol in the given encoding. For example in Shreelipi 5.0 ଙ୍ଗ is mapped to ASCII letter Z which represents ଡ instead in Akruti encoding. Technically speaking these two are encoding over another encoding (ASCII). The ASCII letter Z is itself represented by '5A' in hex, so that is its codepoint.

Although Unicode came into being in 1991 and was supported in Windows NT and Windows 95 onwards (with an update), it was more useful for simpler scripts rather than for complex scripts of Asian languages. When we write କ୍ଷ in Unicode, what actually is written is ' କ୍ ଷ ' because Unicode has only defined base characters in Odia. To convert these combination of base characters into conjuncts (juktakhyara) the Unicode compliant OpenType fonts themselves contain small programs written in a language called OpenType (made jointly by Adobe and Microsoft and later adopted worldwide). All modern OS have something called a rendering engine with layout engine which reads these rules from the font and applies them. Windows 7 and above have DirectWrite, while previous versions have Uniscribe; Mac OS 8.5 to Mac OS X 10.4 used Apple Type Services for Unicode Imaging (ATSUI), the newer versions use Core Text; GNOME based Linux use Pango (With HarfBuzz and FreeType).

This is what is meant as Complex Text Layout or CTL support. Thus while Windows 95 or 98 supported Unicode, they did not have CTL support so Unicode Odia chracters appear but conjuncts will break and pre-base matra (e kar, o kar, au kar) are misplaced. Windows first supported CTL with XP SP2. A huge problem in the past was due to Adobe products not supporting CTL. Although InDesign CS6 first supported it, it was not until the next version (CreativeCloud or CC) that other Adobe products like Photoshop or illustrator started supporting Indic.

Besides, the OpenType program in a font may or may not be complete. In the latter case some conjuncts will break. This is the case with GIST Odia fonts. Similarly, different rendering engines work differently. Thus fonts that work in Windows may not work on Mac (though such cases are rare). InDesign has its own rendering engine to maintain uniformity between Windows and Mac versions. It is much stricter than the others. Thus some fonts that work fine on other apps in Windows, Mac and Linux such as OT_Jagannath or Lohit have problems in InDesign (ମନ୍ମି, ଆର୍ ୱାନ୍).

This all may sound overtly complex and unnecessary while Unicode could have defined all characters including the conjuncts since the codespace has 1,114,112 codepoints, most of which is unused and reserved for future usage. But Unicode's ubiquity makes it the international standard. Besides as an end-user you need not worry about the underlying complexity. You need only worry about the final result which I assure will be just as good (or even better) than if done with any legacy systems.

1.3 Why should you use Unicode?

- ◊ Information exchange: Text in Unicode can be opened on another machine by any Unicode font supporting that language. Every modern OS has at least one Odia Unicode font installed by default (Windows Vista and above – Kalinga, Mac OS X – Sangam Oriya, Linux – Lohit)
- ◊ Backwards compatibility: Your data in Unicode will always be accessible. Text written with earliest versions from 1991 still opens fine.
- ◊ It's free! Not just free of cost, but also free in libre sense. You are no longer tied to something specific to a vendor.
- ◊ Unicode is the standard of web. While you can embed your proprietary fonts for mak-

ing Odia websites, they will not be searchable. Google can't read or understand such text.

◊ Mobile devices: Android will soon come with default Odia font NotoSans. Even now you can root and install an Odia Unicode font on your android to view/enter Odia text.

◊ E-Books: E-books are slowly replacing paper books in the West. As a publisher you might have to publish e-books one day. Using embedded non Unicode fonts is an ugly way and may not even be possible. Best be ready by migrating to Unicode.

For a rough idea why standardization should matter, think of 6-7 yers back when Android mobiles were non-existent. Every company had its own chargers. Nokia had thick and thin pins. Samsung had 3 different types of chargers. LG, Motorola etc every one had different connectors. Now think of the ease the ubiquity of Android has brought in the form of micro-USB chargers. If you forget your charger at home, you can borrow someone else's! micro-USB existed as a standard before Androids came, but Android made it a household presence. I hope InDesign and its support of Unicode with CTL will make the latter standard ubiquitous, eliminating cumbersome legacy encodings.

2. Challenges and Problems for migrating to Unicode

By surveying some professional DTP typesetters and graphic designers in Odia industry I have been able to identify four problems hindering the migration to newer systems. Namely:

1. Use of very old software and hardware
2. Support for old documents
3. Input Method for Unicode Odia
4. Lack of good fonts.

We'll discuss them one by one.

2.1. Use of very old software and hardware

Most DTP typesetters in Odia are using ShreeLipi and Akruti. Though ShreeLipi has an updated version, Akruti does not. Besides many people are using older versions. These do not work on newer Operating System (OS) like Windows 7 or 8. Thus most people are stuck in Windows XP. Some even have backup computers with Windows 98 for tasks like Producing PDF from old copies or opening very old Dos based files.

For publishing purposes the ubiquitous DTP software is Pagemaker. It was made by Aldus in 80 and was already showing its age by mid 90s. It didn't support Unicode or many other features that its rival QuarkXpress did. PM's underlying code was very messy and even its developers had trouble understanding and updating it. That's why Aldus was working on another DTP software which eventually became InDesign. Around the globe PM has mostly vanished but in India it can still be seen widely.

Then there's the problem with hardware. Upgrading hardware is a costly affair but is necessary for running a modern OS.

Solution

Windows NT family support Unicode but CTL (Complex Text Layout) support for Odia came first with Windows XP SP2. So most publishers who are using Windows XP anyway, only need to install a service pack, available from Microsoft.

Adobe's InDesign is the spiritual successor of Pagemaker. It can do anything Pagemaker can do and much more. We recommend InDesign CS6 because it runs on XP, is the first InDesign to support Indic text officially (patches are available for supporting Unicode Indic text from InDesign CS2 and later such as IndicPlus).

As for the hardware, there's no simple solution. You should spend some money on getting at least a Core2Duo with 1GB or more RAM. If you can spend 40K on a printer, you should at least buy new computer. It will cost less than 10K and the investment will be worth it in the long run.

2.2. Support for old documents

Most typesetters and publishers have digital copies of their books in Pagemaker (p65, pmd) format. They must be able to open their old files, edit them or preferably convert them to new format. Also text in other formats such as MS Word files that are encoded in legacy formats like Akruti should ideally be converted to Unicode. Typing all that text again is simply too much.

Solution

You can have Pagemaker 7 installed on Windows XP or 7 for editing your existing Pagemaker files while you can make new ones in InDesign. Even better, InDesign also supports opening PM 6.x-7.x files (only in InDesign CS6 and below. So I don't recommend using InDesign CC if you have lots of old files to support).

I have also made Meghaduta converter which can automatically convert InDesign documents from Akruti/ShreeLipi to Unicode format. See the converter section of this document for details. The converter is based on converters by Wikipedia community which has browser based conversion tool. These browser based tools can be used for converting legacy text in Word or any other files by copy pasting.

2.3. Input Method for Unicode Odia

Inputting Odia text from an English keyboard requires special software called IME (Input Method Editor). ShreeLipi and Akruti packages include their own IME that outputs text in their own encoding and supports several different keyboard layouts. However the lack of a proper Unicode IME for Odia (or rather their obscurity) has prevented typesetters from typing in Unicode. This sometimes results in awkward situations. For example suppose a client supplies a typesetter with a Unicode file using Unicode Kalinga (not to be confused with Akruti Kalinga) font. If the typesetter has no Unicode IME and only has Akruti installed, he'll instead edit the file using Akruti IME and use Akruti Sarala font which resembles Kalinga. This is an ugly solution.

The commonest Keyboard layouts in both ShreeLipi and Akruti for Odia are ShreeLipi Modular layout, Inscript (Govt. of India DoE) and Phonetic (different in both). Most experienced typesetters use Modular and new users use Phonetic. While there are already several Unicode IME for Odia, they have their problems. Most notably they have different keyboard layouts. For someone who is used with a previous layout, learning a new one is less attractive. Two solutions already exist for using Modular layout in Unicode but both don't work 100%.

Solution

If you want to use inScript Layout, it comes built in with Vista and above. If you want to use a layout which doesn't require pre-base placement such as Akruti Phonetic 86 you can easily use MSKLC to create one. It is available here.

However for properly using keyboard layouts like Shreelipi Modular requiring pre and post base placement, you need a powerful IME. KeyMagic IME fills in that gap. I'll discuss how to use it in the IME section of this document.

2.4. Lack of good fonts

This one is more of a myth. There are two dozens of Odia Unicode fonts available for free. For starters Kalinga which is the default Odia font in Vista and above is a very good one for all practical purposes including viewing and printing.

There are several other Unicode fonts: Google's NotoSans Oriya; OT_Jagannatha and others designed by Sujata Patel; Odia fonts from GIST etc.

2.5. Prerequisites

◊ Before using InDesign you must have at least ServicePack3 (SP3) update installed for Win XP. It can be obtained from the Microsoft site: https://www.microsoft.com/en-us/download/details.aspx?id=24

(Choose 'Proceed with this download' -> 'No Thanks and Continue')

◊ KeyMagic requires Microsoft Visual C Redist 2008 and Microsoft .NET framework 3.5 installed. They can be obtained from

.NET 3.5: http://www.microsoft.com/download/en/details.aspx?id=21

vc2008 redist: http://www.microsoft.com/download/en/details.aspx?id=5582

3. Current Projects

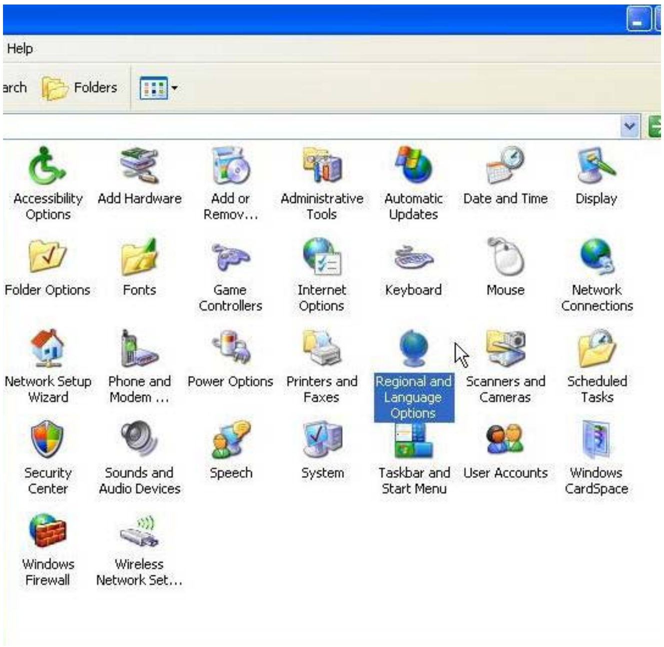
3.1. IME (Kunji-Binyasa Project)

For inputting Odia text deciding what Keyboard layout you need is important.

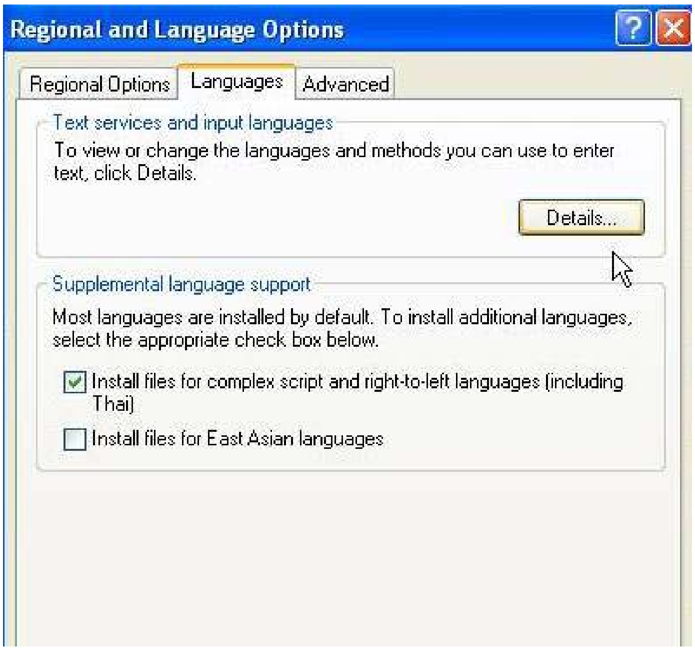
3.1.1. Inscript

- ◊ Windows XP: Download and install Inscript keybord from here. Download the zip file by right click -> save link as. Extract and run setup.exe. Now go to Control Panel -> Regional and language options -> select the languages tab -> Details -> Add -> Select any other language since Oriya is not listed here -> From 2nd dropdown menu select Inscript Layout for Odia -> OK -> Apply -> OK
- ◊ Windows Vista and above: Built in. To enable go to Control Panel -> Clock, Language, and Region -> Change keyboards or other input methods -> Change keyboards -> Add -> Oriya(India) -> Keyboard, now check Oriya -> OK -> Apply.

Note: The inbuilt Inscript Keyboard layout has multiple errors such as absence of purnacched, no Odia numerals, ଯ଼ instead of ଯ and ଯ instead of ୟ, ଵ instead of ୱ as tested on Win 7. I recommend using the above link or KeyMagic below.



[Control Panel -> Regional and Language Options]


Regional and Language Options

Regional Options | Languages | Advanced

Text services and input languages

To view or change the languages and methods you can use to enter text, click Details.

Details...

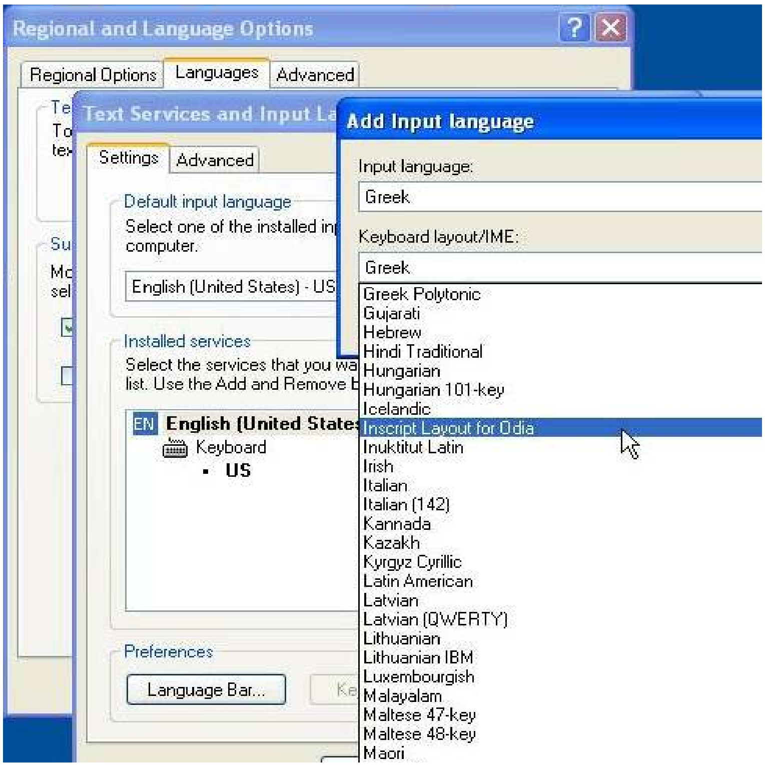
Supplemental language support

Most languages are installed by default. To install additional languages, select the appropriate check box below.

☑ Install files for complex script and right-to-left languages (including Thai)

□ Install files for East Asian languages

[Languages -> Details]



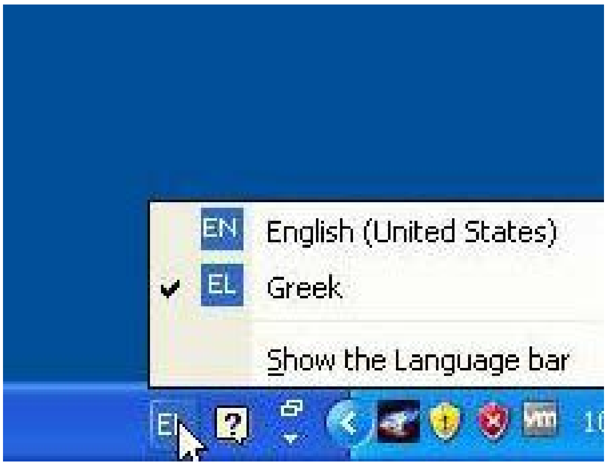
English (United States) - US

Input language: Greek

Keyboard layout/IME: Greek

[Left]
[Add -> Select any other language -> Select ‘Inscript Layout for Odia’ from 2nd menu -> OK -> Apply -> OK]

[Below]
[Select the language from the language pane.]



3.1.2. ShreeLipi Modular

1. First download KeyMagic for your platform from here (For Windows choose the 4th one).
2. Now download the layout file from here
3. Intall KeyMagic and then launch it.
4. Click on ‘Add’ on the right side, browse to where you downloaded the km2 file and select it.
5. Double click on the new entry and assign a hotkey. Use something easy to remember.
6. Press the Hotkey to switch to Odia and press it again to switch back to English. You can also seamlessly switch between multiple layouts. Kunji-Binyasa project offers other Keyboard layouts too. You can download the appropriate km2 file and load it in KeyMagic. See what other layouts are available here.

3.1.3 Phonetic and other layouts

◊ Kunji Binyasa project offers Akruti Phonetic 86 layout for use with KeyMagic.

◊ Microsoft ILIT allows you to type in English and transliterate it into Odia. You can also use ILIT offline.

◊ TypeOdia (click download zip, then extract and open the HTML file in firefox browser). It offers several different layouts.

◊ Several other open as well as proprietary Unicode IME are also available such as Lekha Odia, Baraha IME, dhwanI Unicode, Apranta etc.

3.2. Converter (Meghaduta Project)

3.1.1 Introduction

Most DTP users have files in Pagemaker format. These can be converted using the browser based converters made by Odia Wikipedia community by copy pasting. For hundreds of pages of a book, this is a very tedious process.

Since InDesign can open PM files and supports scripting, I have made two scripts for InDesign that can automatically convert all text in a document to Unicode format: one each for ShreeLipi v4/5/6 and Akruti. Two more converters are being made for Old ShreeLipi (v3.0) and ShreeLipiEx (v7.0). They have been tested on XP and Win 7 using iD CS6. They should work in newer versions of InDesign as well.

3.2.2. How To Use

- ◊ Download Meghaduta Converter. Right click on Meghaduta.exe -> save link as then install it.
- ◊ Open your PM/iD document in InDesign. Go to Window -> Utilities -> Scripts. You should see a folder called Unicode Converters under user in scripts panel.
- ◊ Open the plain or smart folder and run appropriate converter by double clicking. If your document is big it can take some time.

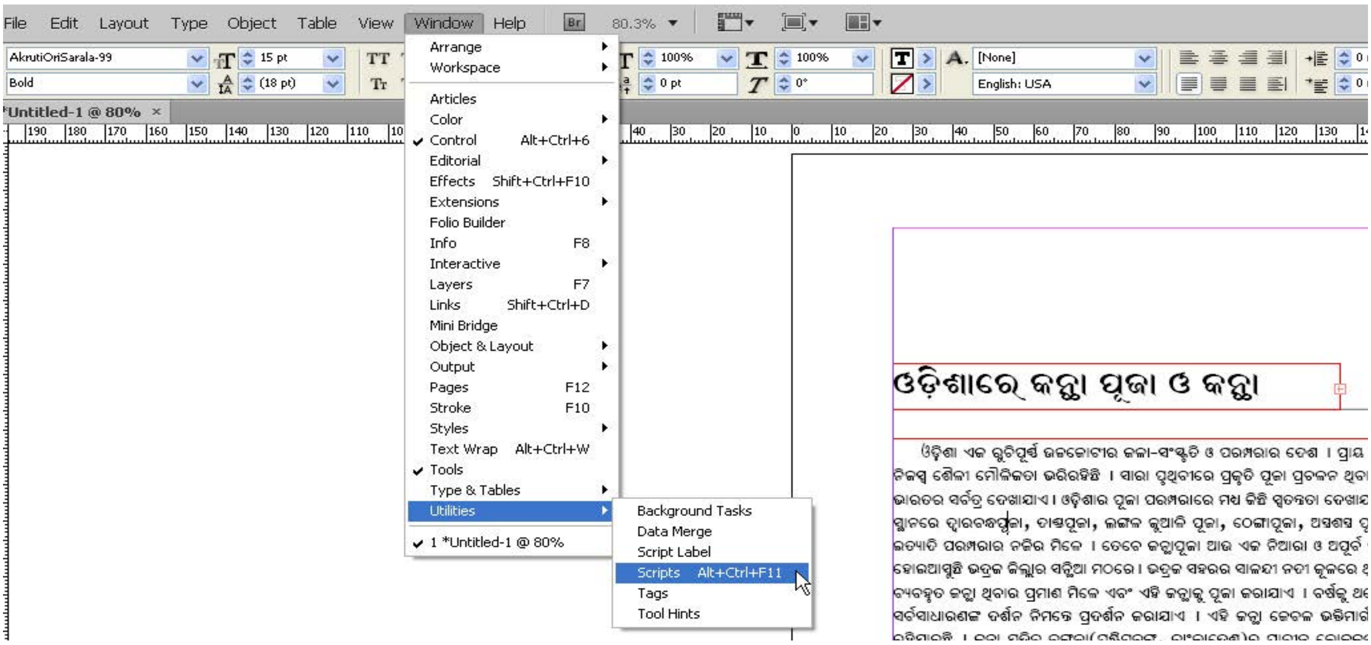
3.2.3 Plain vs. Smart converter

Plain script converts all the text (except footers and headers). Since Latin letters represent Odia letters in Akruti/ShreeLipi encoding, so if a documents mixes Odia with English text, the English text is also force-converted resulting in garbled output.

Smart script converts text formatted in a particular font. So if you have Odia text in Akruti Sarala mixed with English text in Times New Roman, you can choose to convert only the text in Akruti Sarala and the English text is spared.

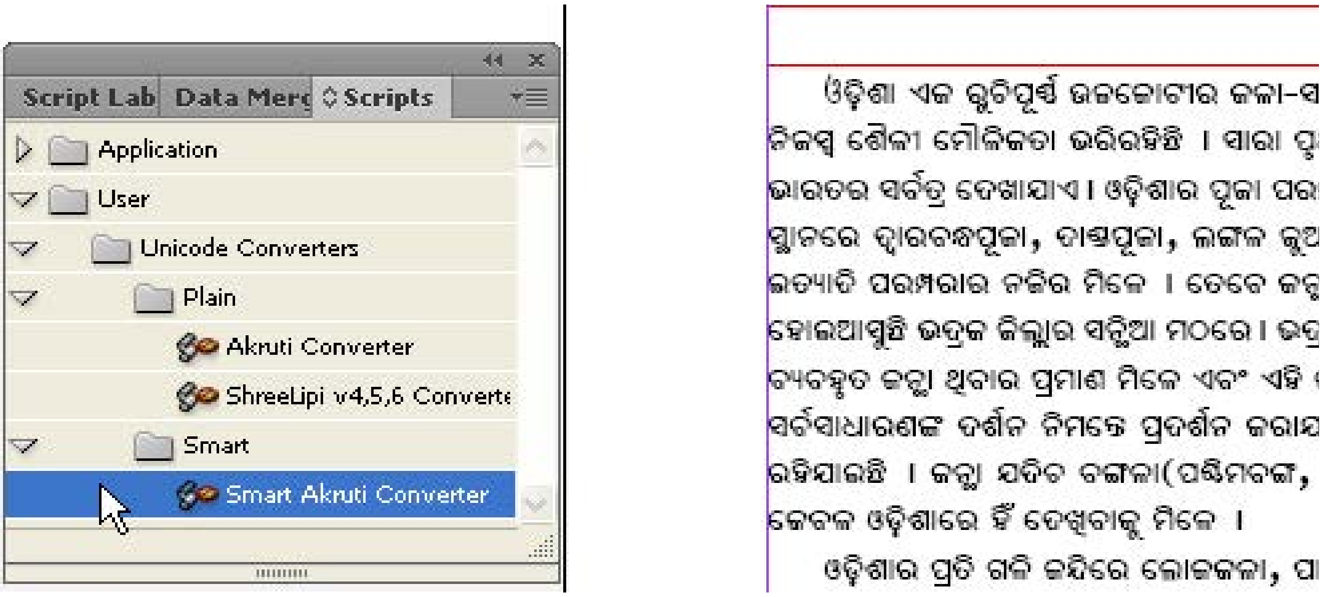
Smart Script is also useful in another scenario: to apply styles if you have not applied styles and manually formatted all the text. Say you have a document with body text in Akruti Sarala and headings in Akruti Konarka. With Smart script you can choose to convert Sarala text to Kalinga. Then convert Konarka text to Kalinga bold.

There's a known problem with InDesign where it cannot properly open certain ShreeLipi encoded Pagemaker files. It will refuse to find the used ShreeLipi font and automatically use another font. Smart converter will fail to work here. You have to use the plain one. Besides InDesign garbles some letters while opening some troublesome ShreeLipi 3.0 files so the converter may not work properly. This is a problem with InDesign, I hope to make a separate script for these soon. Until then, hang on to PM for these files or use the browser based converter by copying from Pagemaker and pasting into InDesign.



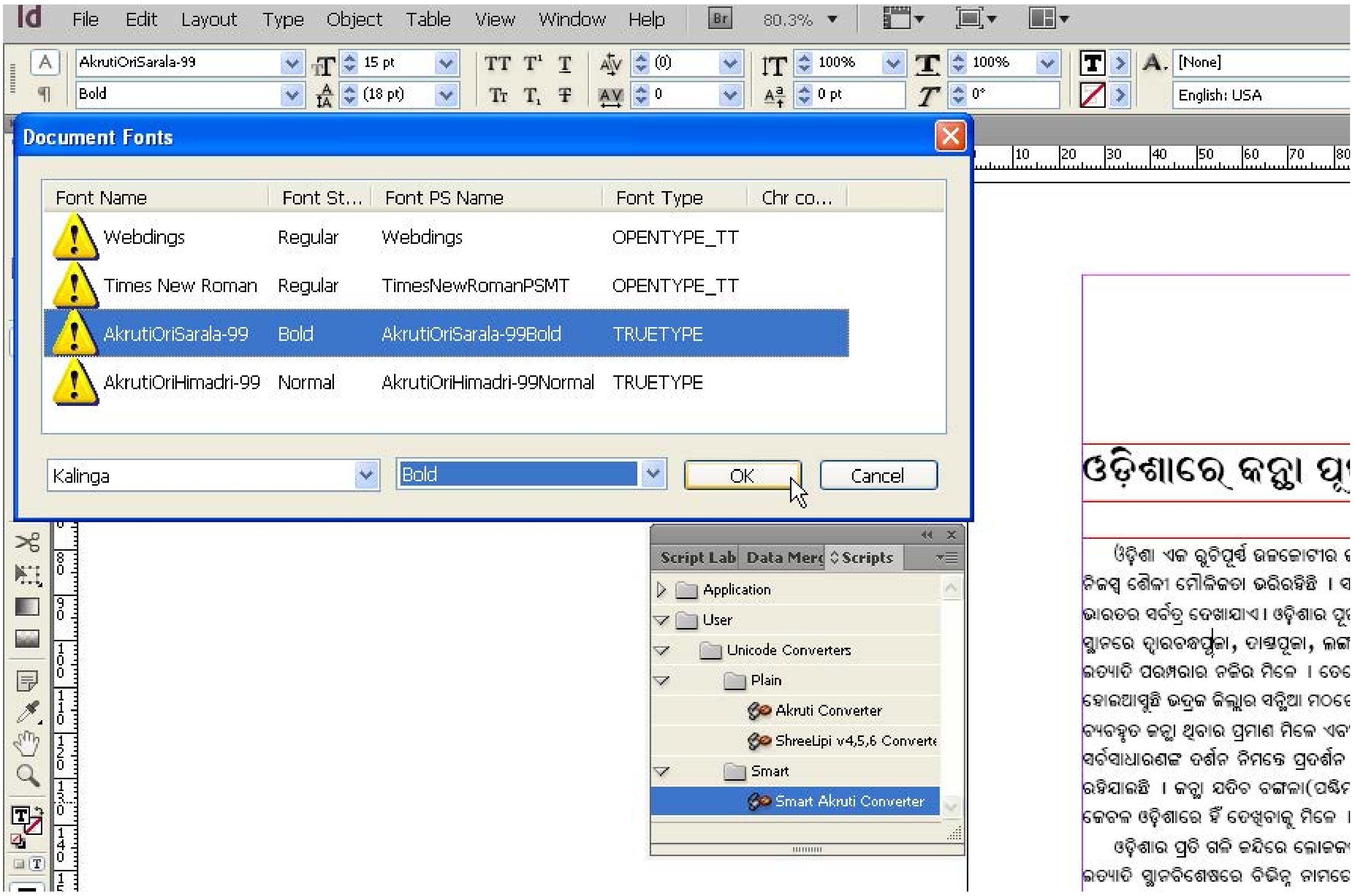
ଓଡ଼ିଶାରେ କନ୍ଥା ପୂଜା ଓ କନ୍ଥା

[First launch the scripts panel]



ଓଡ଼ିଶା ଏକ ଋଦ୍ଧିପୂର୍ଣ୍ଣ ଉତ୍କୋଟୀର କଳା-ସ
ନିଜସ୍ୱ ଶୈଳୀ ମୌଳିକତା ଭରିରହିଛି । ସାରା ପୃ
ଭାରତର ସର୍ବତ୍ର ଦେଖାଯାଏ । ଓଡ଼ିଶାର ପୂଜା ପର
ସ୍ଥାନରେ ଦ୍ୱାରବନ୍ଧପୂଜା, ଡାଣ୍ଡପୂଜା, ଲଙ୍ଗଳ କୁଅ
ଇତ୍ୟାଦି ପରମ୍ପରାର ନଜିର ମିଳେ । ତେବେ କନ୍ଥ
ହୋଇଆସୁଛି ଭଦ୍ରକ ଜିଲ୍ଲାର ସନ୍ଥିଆ ମଠରେ । ଭଦ୍
ବ୍ୟବହୃତ କନ୍ଥା ଥିବାର ପ୍ରମାଣ ମିଳେ ଏବଂ ଏହି
ସର୍ବସାଧାରଣଙ୍କ ଦର୍ଶନ ନିମନ୍ତେ ପ୍ରଦର୍ଶନ କରାଯ
ରହିଯାଇଛି । କନ୍ଥା ଯଦିଓ ବଙ୍ଗଳା(ପଶ୍ଚିମବଙ୍ଗ,
କେବଳ ଓଡ଼ିଶାରେ ହିଁ ଦେଖିବାକୁ ମିଳେ ।
ଓଡ଼ିଶାର ପ୍ରତି ଗଳି କନ୍ଦିରେ ଲୋକକଳା, ପ

[Run a script by double clicking on it in the scripts panel. Scripts can be grouped in folders as seen here]

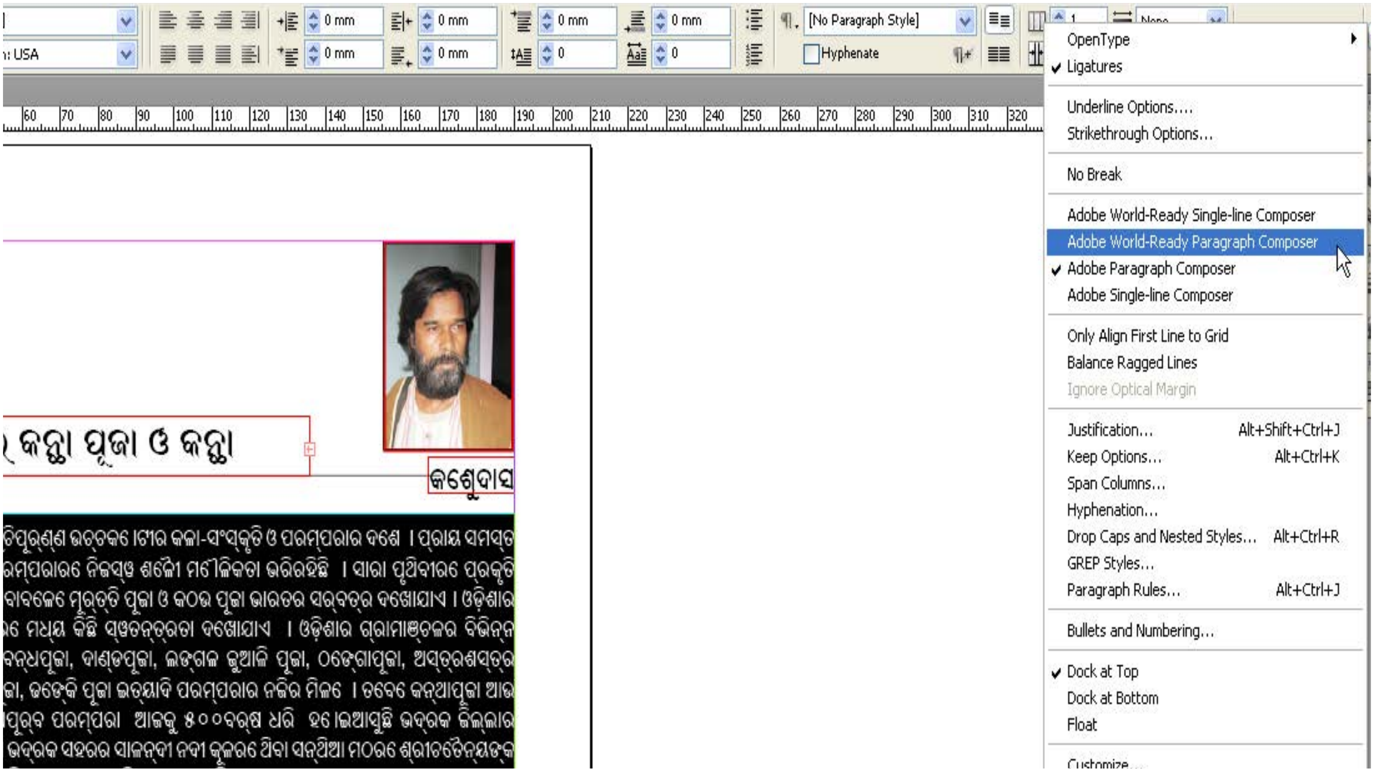


Document Fonts

Font Name	Font St...	Font PS Name	Font Type	Chr co...
Webdings	Regular	Webdings	OPENTYPE_TT	
Times New Roman	Regular	TimesNewRomanPSMT	OPENTYPE_TT	
AkrutiOriSarala-99	Bold	AkrutiOriSarala-99Bold	TRUETYPE	
AkrutiOriHimadri-99	Normal	AkrutiOriHimadri-99Normal	TRUETYPE	

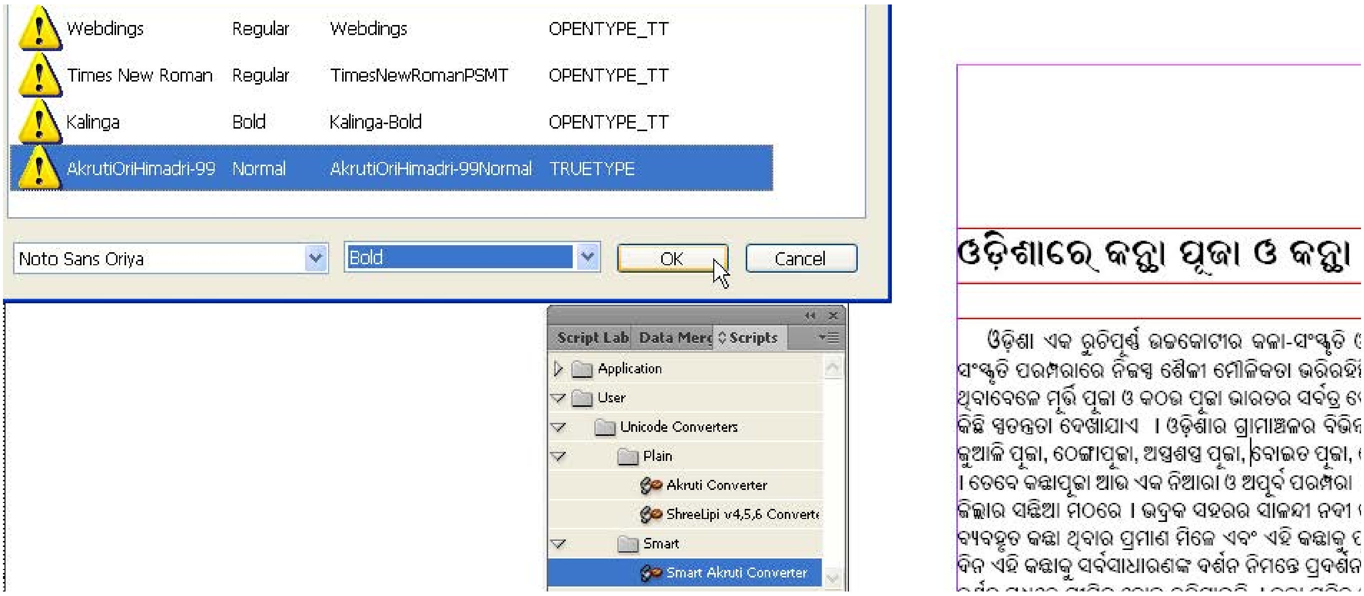
Kalinga · Bold

[Dialogue box of smart converter which can detect which fonts are used in the document. We choose to convert only the text in Akruti Sarala (body text) to Kalinga Unicode (Bold)]



[No Paragraph Style]
Hyphenate
OpenType
✓ Ligatures
Underline Options....
Strikethrough Options...
No Break
Adobe World-Ready Single-line Composer
Adobe World-Ready Paragraph Composer
✓ Adobe Paragraph Composer
Adobe Single-line Composer
Only Align First Line to Grid
Balance Ragged Lines
Ignore Optical Margin
Justification... Alt+Shift+Ctrl+J
Keep Options... Alt+Ctrl+K
Span Columns...
Hyphenation...
Drop Caps and Nested Styles... Alt+Ctrl+R
GREP Styles...
Paragraph Rules... Alt+Ctrl+J
Bullets and Numbering...
✓ Dock at Top
Dock at Bottom
Float
Customize...

[You can notice the body text is converted to Unicode but is broken. This can be fixed by choosing ‘Adobe World-Ready Paragraph composer’ from top right]



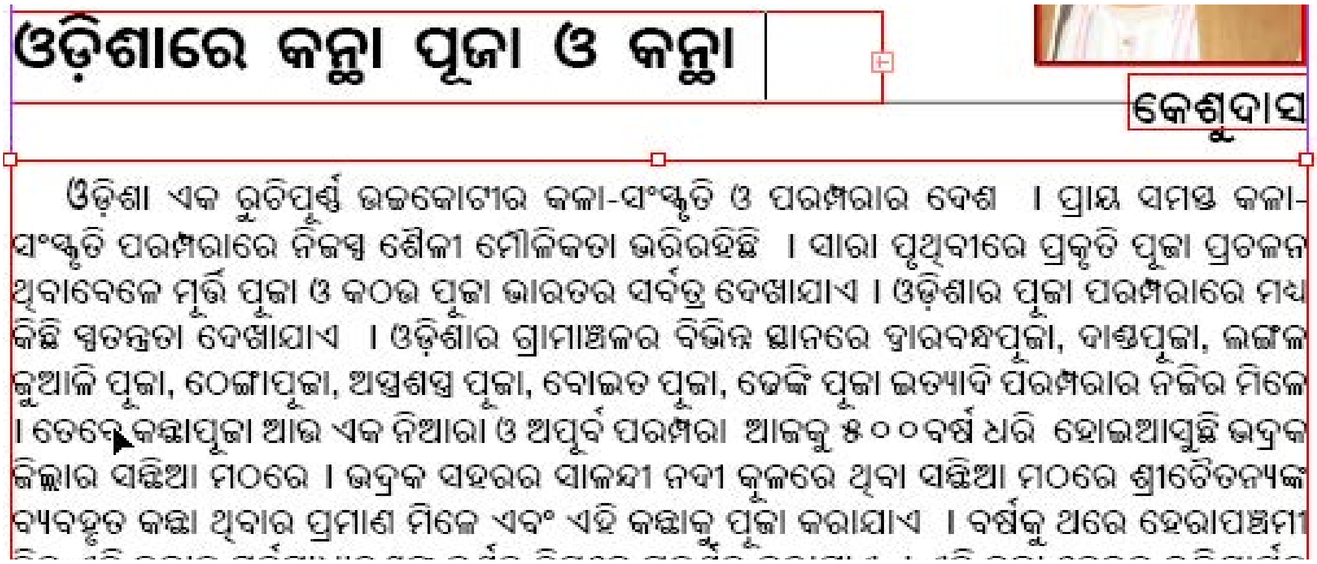
Webdings	Regular	Webdings	OPENTYPE_TT
Times New Roman	Regular	TimesNewRomanPSMT	OPENTYPE_TT
Kalinga	Bold	Kalinga-Bold	OPENTYPE_TT
AkrutiOriHimadri-99	Normal	AkrutiOriHimadri-99Normal	TRUETYPE

Noto Sans Oriya | Bold

ଓଡ଼ିଶାରେ କନ୍ଥା ପୂଜା ଓ କନ୍ଥା

ଓଡ଼ିଶା ଏକ ରୁଚିପୂର୍ଣ୍ଣ ଉଚ୍ଚକୋଟୀର କଳା-ସଂସ୍କୃତି
ସଂସ୍କୃତି ପରମ୍ପରାରେ ନିଜସ୍ୱ ଶୈଳୀ ମୌଳିକତା ଭରିରହି
ଥିବାବେଳେ ମୂର୍ତ୍ତି ପୂଜା ଓ କଠଉ ପୂଜା ଭାରତର ସର୍ବତ୍ର
କିଛି ସ୍ୱତନ୍ତ୍ରତା ଦେଖାଯାଏ । ଓଡ଼ିଶାର ଗ୍ରାମାଞ୍ଚଳର ବିଭି
କୁଆଳି ପୂଜା, ଠେଙ୍ଗାପୂଜା, ଅସ୍ତ୍ରଶସ୍ତ୍ର ପୂଜା, ବୋଇତ ପୂଜା,
। ତେବେ କନ୍ଥାପୂଜା ଆଉ ଏକ ନିଆରା ଓ ଅପୂର୍ବ ପରମ୍ପରା
ଜିଲ୍ଲାର ସଣ୍ଢିଆ ମଠରେ । ଭଦ୍ରକ ସହରର ସାଳନ୍ଦୀ ନଦୀ
ବ୍ୟବହୃତ କନ୍ଥା ଥିବାର ପ୍ରମାଣ ମିଳେ ଏବଂ ଏହି କନ୍ଥାକୁ
ଦିନ ଏହି କନ୍ଥାକୁ ସର୍ବସାଧାରଣଙ୍କ ଦର୍ଶନ ନିମନ୍ତେ ପ୍ରଦର୍ଶନ

[Notice body text is fixed now. Now we convert the heading in AK Himadri to Noto Sans (Bold)]

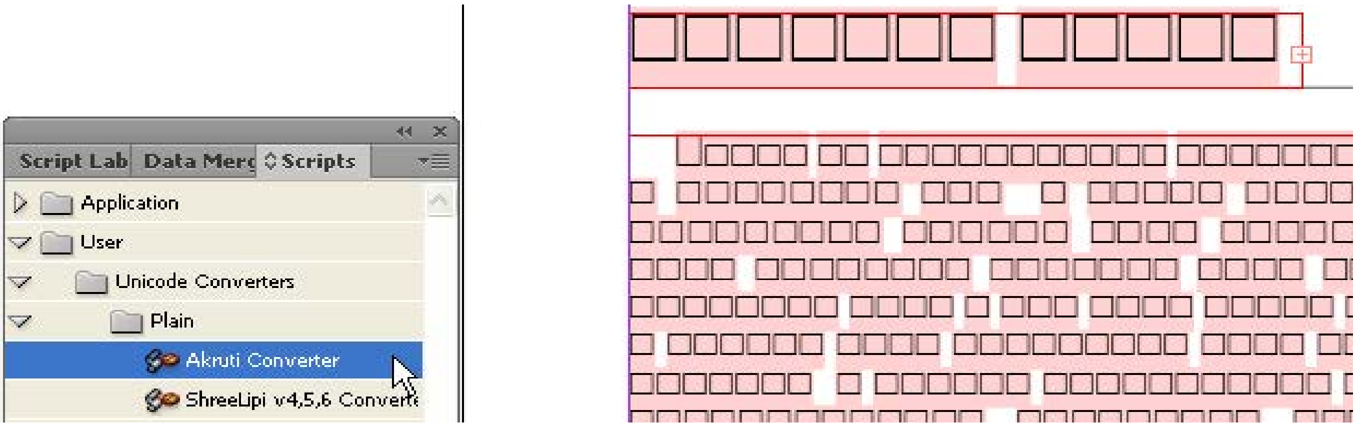


ଓଡ଼ିଶାରେ କନ୍ଥା ପୂଜା ଓ କନ୍ଥା

କେଶୁଦାସ

ଓଡ଼ିଶା ଏକ ରୁଚିପୂର୍ଣ୍ଣ ଉଚ୍ଚକୋଟୀର କଳା-ସଂସ୍କୃତି ଓ ପରମ୍ପରାର ଦେଶ । ପ୍ରାୟ ସମସ୍ତ କଳା-ସଂସ୍କୃତି ପରମ୍ପରାରେ ନିଜସ୍ୱ ଶୈଳୀ ମୌଳିକତା ଭରିରହିଛି । ସାରା ପୃଥିବୀରେ ପ୍ରକୃତି ପୂଜା ପ୍ରଚଳନ ଥିବାବେଳେ ମୂର୍ତ୍ତି ପୂଜା ଓ କଠଉ ପୂଜା ଭାରତର ସର୍ବତ୍ର ଦେଖାଯାଏ । ଓଡ଼ିଶାର ପୂଜା ପରମ୍ପରାରେ ମଧ୍ୟ କିଛି ସ୍ୱତନ୍ତ୍ରତା ଦେଖାଯାଏ । ଓଡ଼ିଶାର ଗ୍ରାମାଞ୍ଚଳର ବିଭିନ୍ନ ସ୍ଥାନରେ ଦ୍ୱାରବନ୍ଧପୂଜା, ଦାଣ୍ଡପୂଜା, ଲଙ୍ଗଳ କୁଆଳି ପୂଜା, ଠେଙ୍ଗାପୂଜା, ଅସ୍ତ୍ରଶସ୍ତ୍ର ପୂଜା, ବୋଇତ ପୂଜା, ଢେଙ୍କି ପୂଜା ଇତ୍ୟାଦି ପରମ୍ପରାର ନଜିର ମିଳେ । ତେବେ କନ୍ଥାପୂଜା ଆଉ ଏକ ନିଆରା ଓ ଅପୂର୍ବ ପରମ୍ପରା ଆଜକୁ ୫୦୦ ବର୍ଷ ଧରି ହୋଇଆସୁଛି ଭଦ୍ରକ ଜିଲ୍ଲାର ସଣ୍ଢିଆ ମଠରେ । ଭଦ୍ରକ ସହରର ସାଳନ୍ଦୀ ନଦୀ କୂଳରେ ଥିବା ସଣ୍ଢିଆ ମଠରେ ଶ୍ରୀଚୈତନ୍ୟଙ୍କ ବ୍ୟବହୃତ କନ୍ଥା ଥିବାର ପ୍ରମାଣ ମିଳେ ଏବଂ ଏହି କନ୍ଥାକୁ ପୂଜା କରାଯାଏ । ବର୍ଷକୁ ଥରେ ହେରାପଞ୍ଚମୀ

[The final result]



[The result of running plain converter on the same source file. It indiscriminately converts everything and also doesn't apply Unicode font which must be done manually]

3.3. Fonts (Lekhani Project)

3.3.1 Introduction

As I mentioned before there are about two dozen Odia Unicode fonts. While available variety is less, this is more of a problem for graphic designers than for DTP typesetters who mostly use 2-4 different fonts at most. Kalinga can be used for body text instead of Akruti Sarala/ Lingaraj or Shree Ori 601. Similarly Kalinga Bold can replace Akruti Konarka or Shree Ori 602 for headings. OT Jagannath can be used for body text also. Its rounded letters and combined i kar give it a beautiful authentic old print style. NotoSans can be used for slightly stylized and modern look. Baloo Bhaina has a playful and bold look. If you require more styles GIST fonts or Sujata Patel's fonts are also available for free non-commercial use. (PS: GIST fonts cannot be embedded in PDF)

So what exactly is Lekhani project? It is for producing several high quality stylized Odia fonts under open source license. Currently only OT_Jagannatha, NotoSans, Sakal Bharati and Lohit Oriya are open source (OFL). The project currently has no new fonts. Lekhani base is a modification of NotoSans. You can help me by providing artwork for Oriya letters and I will convert them into a font. If you want to make a font yourself I will soon make a detailed tutorial on making a Unicode compliant OpenType Odia font from scratch.

3.3.2 Currently available Odia Unicode fonts

◊ Kalinga: (ସଙ୍ଗୀତ ମୂର୍ଚ୍ଛନାର ତାଳେ ନାଚି) One of the best quality Odia font. Made by Microsoft. Bold version available. Bundled with Vista and above.

◊ OT_Jagannatha: (ସଙ୍ଗୀତ ମୂର୍ଚ୍ଛନାର ତାଳେ ନାଚି) Rounded old style text. Made by Sujata Patel. Bold not available. http://www.odialanguage.com/Odia_fonts.html

◊ NotoSans: (ସଙ୍ଗୀତ ମୂର୍ଚ୍ଛନାର ତାଳେ ନାଚି) Modern typeface by Google for use in Android. Bold available. https://www.google.com/get/noto/#sans-orya

◊ Baloo Bhaina: (**ସଙ୍ଗୀତ ମୂର୍ଚ୍ଛନାର ତାଳେ ନାଚି**) Heavy display typeface by the Ek Type Foundry. https://github.com/girish-dalvi/Baloo

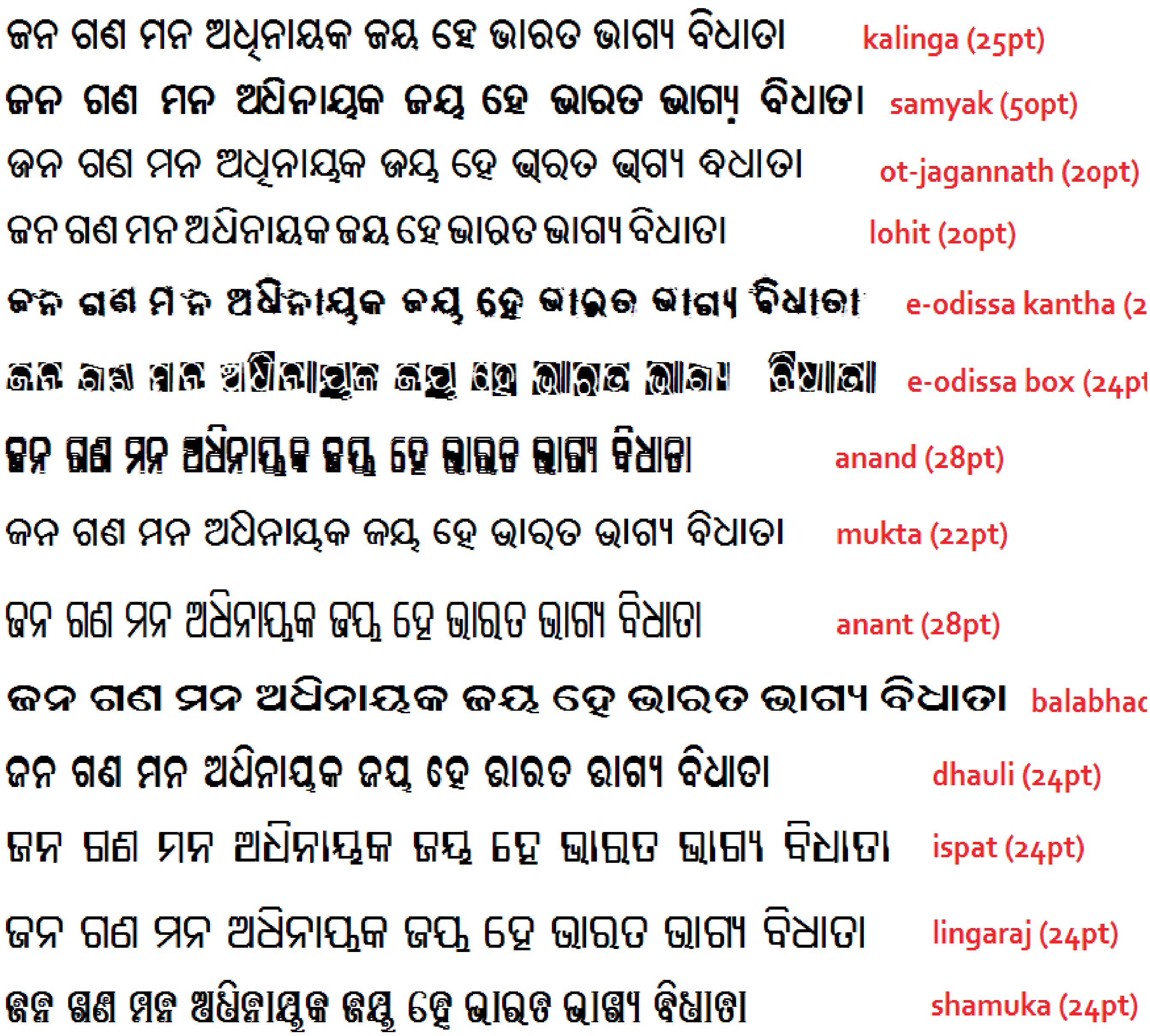
◊ Sakal Bharati: (ସଙ୍ଗୀତ ମୂର୍ଚ୍ଛନାର ତାଳେ ନାଚି) One font supporting all Indian languages. Made by GIST, C-DAC. http://cdac.in/index.aspx?id=dl_SakalBharati_Ship

◊ GIST Fonts: Catalogue is available.

◊ Sujata Patel's fonts: Several stylish typefaces are available. http://odiafonts.com/top_rated/

◊ Other fonts: There are several other Unicode fonts available too. You can check out Luc Devroye' s Odia page.

Note: All of these fonts can be used in any Unicode compliant software supporting CTL like MS Word. However InDesign uses its own shaper so some fonts might not work well. Most notably OT_Jagannath and Lohit Oriya have problems with ra phalaa, ma phalaa and a few others. This is because they use old OT code. I've fixed the problem with OT_Jagannath so that it works in InDesign. It is currently in testing phase, if you need to use it in InDesign, mail me and I'll send it to you.



ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା kalinga (25pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା samyak (50pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭ୍ରତ ଭ୍ଗ୍ୟ ବଧାତା ot-jagannath (20pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା lohit (20pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା e-odissa kantha (2

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା e-odissa box (24pt

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା anand (28pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା mukta (22pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା anant (28pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା balabhac

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା dhauli (24pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା ispat (24pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା lingaraj (24pt)

ଜନ ଗଣ ମନ ଅଧିନାୟକ ଜୟ ହେ ଭାରତ ଭାଗ୍ୟ ବିଧାତା shamuka (24pt)

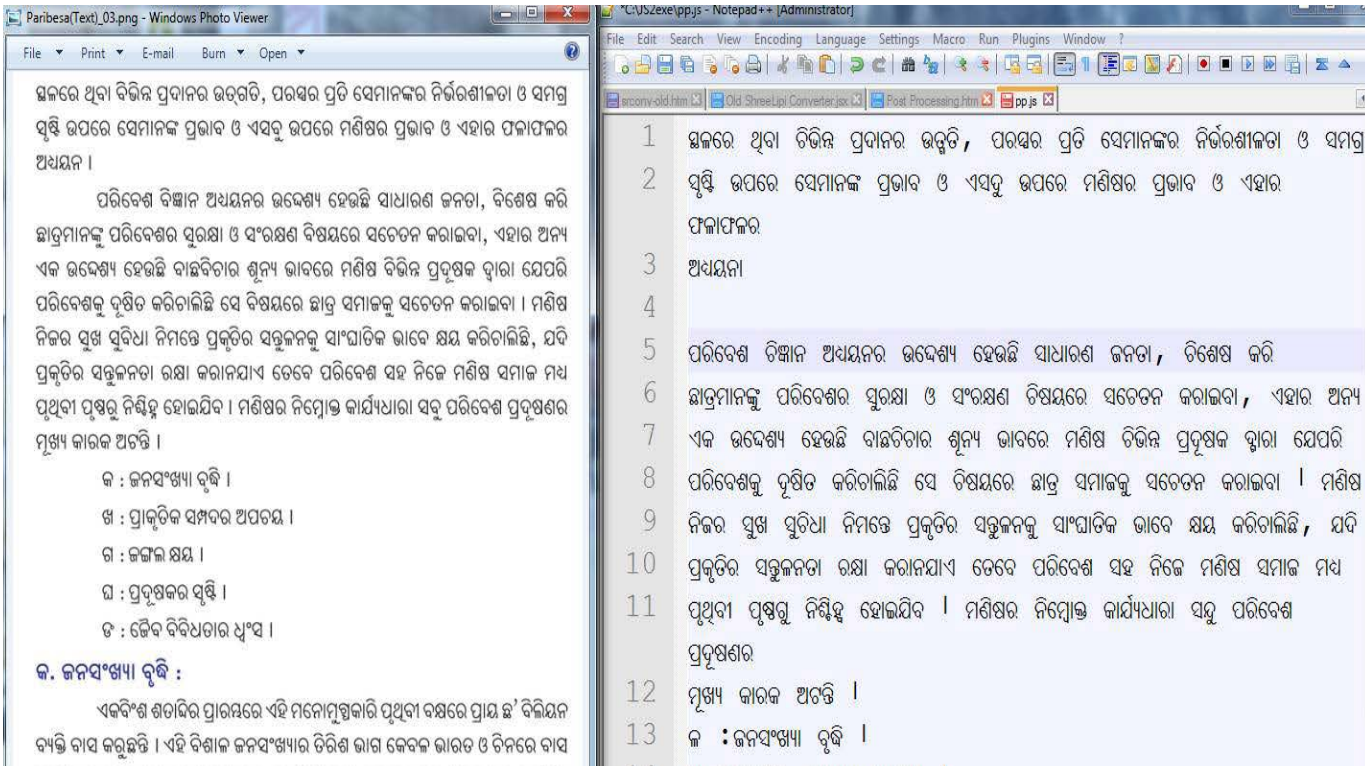
[A sample of some available unicode Odia fonts]

3.4. OCR (Project Pratilipi)

I'm currently working on training Tesseract OCR software for recognizing and digitising printed material. The original aim of the project is to digitise old public domain literature in Odia for hosting on Odia Wikisource. However even typesetters and Publishers can also leverage this. Very often they get printed manuscripts of previous edition of a book without a digital copy. These have to be typed in again. An OCR can be a real timesaver in such cases.

The project is currently nascent. OCR is highly dependent on the typeface used in the print. The most common typeface used in Odia printing since 90s is what I call "Modern Odia Typeface". Shree 601, Akruti Sarala, Kalinga, Lohit are examples of such. My OCR currently recognizes this typeface from image files made on computer (screenshots, PDF files with unknown encoding) with 80-90% accuracy. With very clean scans of printed text this is about 70-80%. With less clean scans with warped text or old greyed books this drops further.

The only image pre-processing I do is with Scantailor. A further improvement in that along with the production of an Odia spellchecker can improve detection by an extra 5-15%. I've only trained it with computer generated images. I do not have a scanner so I'm restricted to work on scans available on internet. Most of what I could find were low resolution images. With extensive training, accuracy can be boosted to 85-92% for most scans and 99% for generated images.


Paribesa(Text)_03.png - Windows Photo Viewer
File
Print
E-mail
Burn
Open
*C:\JS2exe\pp.js - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
srconv-old.htm
Old ShreeLipi Converter.jsx
Post Processing.htm
pp.js

ସ୍ଥଳରେ ଥିବା ବିଭିନ୍ନ ପ୍ରଦାନର ଉତ୍‌ଗତି, ପରସ୍ପର ପ୍ରତି ସେମାନଙ୍କର ନିର୍ଭରଶୀଳତା ଓ ସମଗ୍ର ସୃଷ୍ଟି ଉପରେ ସେମାନଙ୍କ ପ୍ରଭାବ ଓ ଏସବୁ ଉପରେ ମଣିଷର ପ୍ରଭାବ ଓ ଏହାର ଫଳାଫଳର ଅଧ୍ୟୟନ ।

ପରିବେଶ ବିଜ୍ଞାନ ଅଧ୍ୟୟନର ଉଦ୍ଦେଶ୍ୟ ହେଉଛି ସାଧାରଣ ଜନତା, ବିଶେଷ କରି ଛାତ୍ରମାନଙ୍କୁ ପରିବେଶର ସୁରକ୍ଷା ଓ ସଂରକ୍ଷଣ ବିଷୟରେ ସଚେତନ କରାଇବା, ଏହାର ଅନ୍ୟ ଏକ ଉଦ୍ଦେଶ୍ୟ ହେଉଛି ବାଛବିଚାର ଶୂନ୍ୟ ଭାବରେ ମଣିଷ ବିଭିନ୍ନ ପ୍ରଦୂଷକ ଦ୍ୱାରା ଯେପରି ପରିବେଶକୁ ଦୂଷିତ କରିଚାଲିଛି ସେ ବିଷୟରେ ଛାତ୍ର ସମାଜକୁ ସଚେତନ କରାଇବା । ମଣିଷ ନିଜର ସୁଖ ସୁବିଧା ନିମନ୍ତେ ପ୍ରକୃତିର ସନ୍ତୁଳନକୁ ସାଂଘାତିକ ଭାବେ କ୍ଷୟ କରିଚାଲିଛି, ଯଦି ପ୍ରକୃତିର ସନ୍ତୁଳନତା ରକ୍ଷା କରାନଯାଏ ତେବେ ପରିବେଶ ସହ ନିଜେ ମଣିଷ ସମାଜ ମଧ ପୃଥିବୀ ପୃଷ୍ଠରୁ ନିଶ୍ଚିହ୍ନ ହୋଇଯିବ । ମଣିଷର ନିମ୍ନୋକ୍ତ କାର୍ଯ୍ୟଧାରା ସବୁ ପରିବେଶ ପ୍ରଦୂଷଣର ମୂଖ୍ୟ କାରକ ଅଟନ୍ତି ।

କ : ଜନସଂଖ୍ୟା ବୃଦ୍ଧି ।

ଖ : ପ୍ରାକୃତିକ ସମ୍ପଦର ଅପଚୟ ।

ଗ : ଜଙ୍ଗଲ କ୍ଷୟ ।

ଘ : ପ୍ରଦୂଷକର ସୃଷ୍ଟି ।

ଙ : ଜୈବ ବିବିଧତାର ଧ୍ୱଂସ ।

କ. ଜନସଂଖ୍ୟା ବୃଦ୍ଧି :

ଏକବିଂଶ ଶତାବ୍ଦିର ପ୍ରାରମ୍ଭରେ ଏହି ମନୋମୁଗ୍ଧକାରି ପୃଥିବୀ ବକ୍ଷରେ ପ୍ରାୟ ଛ' ବିଲିୟନ ବ୍ୟକ୍ତି ବାସ କରୁଛନ୍ତି । ଏହି ବିଶାଳ ଜନସଂଖ୍ୟାର ତିରିଶ ଭାଗ କେବଳ ଭାରତ ଓ ଚିନରେ ବାସ

```
ସ୍ଥଳରେ ଥିବା ଚିଭିନ୍ନ ପ୍ରଦାନର ଉଦ୍ଗତି, ପରସ୍ପର ପ୍ରତି ସେମାନଙ୍କର ନିର୍ଭରଶୀଳତା ଓ ସମଗ୍ର
ସୃଷ୍ଟି ଉପରେ ସେମାନଙ୍କ ପ୍ରଭାବ ଓ ଏସଦୁ ଉପରେ ମଣିଷର ପ୍ରଭାବ ଓ ଏହାର
ଫଳାଫଳର
ଅଧ୍ୟୟନା

ପରିବେଶ ଚିଜ୍ଞାନ ଅଧ୍ୟୟନର ଉଦ୍ଦେଶ୍ୟ ହେଉଛି ସାଧାରଣ ଜନତା, ଚିଶେଷ କରି
ଛାତ୍ରମାନଙ୍କୁ ପରିବେଶର ସୁରକ୍ଷା ଓ ସଂରକ୍ଷଣ ଚିଷୟରେ ସଚେତନ କରାଇବା, ଏହାର ଅନ୍ୟ
ଏକ ଉଦ୍ଦେଶ୍ୟ ହେଉଛି ବାଛଚିଚାର ଶୂନ୍ୟ ଭାବରେ ମଣିଷ ଚିଭିନ୍ନ ପ୍ରଦୂଷକ ଦ୍ବାରା ଯେପରି
ପରିବେଶକୁ ଦୂଷିତ କରିଚାଲିଛି ସେ ଚିଷୟରେ ଛାତ୍ର ସମାଜକୁ ସଚେତନ କରାଇବା । ମଣିଷ
ନିଜର ସୁଖ ସୁଚିଧା ନିମନ୍ତେ ପ୍ରକୃତିର ସନ୍ତୁଳନକୁ ସାଂଘାତିକ ଭାବେ କ୍ଷୟ କରିଚାଲିଛି, ଯଦି
ପ୍ରକୃତିର ସନ୍ତୁଳନତା ରକ୍ଷା କରାନଯାଏ ତେବେ ପରିବେଶ ସହ ନିଜେ ମଣିଷ ସମାଜ ମଧ
ପୃଥିବୀ ପୃଷ୍ଠଗୁ ନିଶ୍ଚିହ୍ନ ହୋଇଯିବ । ମଣିଷର ନିମ୍ନୋକ୍ତ କାର୍ଯ୍ୟଧାରା ସନ୍ଦୁ ପରିବେଶ
ପ୍ରଦୂଷଣର
ମୂଖ୍ୟ କାରକ ଅଟନ୍ତି ।
ଳ :ଜନସଂଖ୍ୟା ବୃଦ୍ଧି ।
```

[Raw OCR output (right) from a page extracted from PDF as jpg file (left)]

4. Acknowledgements

Thank you Wikipedia community. You people spurred my interest in working for Odia language. Particularly Subhashish Panigrahi, Mrutyunjaya Kar, Jnanaranjan Sahoo and Shitikantha Das. Thank you for your encouragements and support, both general and technical.

Thanks to Srujanika who pioneered the work on bringing Odia to digital frontier. My Meghaduta converter is based on Odia Wikipedia's converter which itself is inspired from Rebati converter by Srujanika.

Thanks also goes to Swami Shivadhyanananda of RK Mission, Bibhu Sahoo of Graph and Graphics and Md. Riaz of Unicorn graphics for agreeing to provide me with source files, test my programs and also report bugs.

I'm also thankful to Manoj babu of Grantha Mandir for his interest in Unicode, for putting me in touch with the people in the print industry and rekindling my interest in taking up this project.

“No power on earth can stop an idea whose time has come.”

—Victor Hugo (quoted by Dr. Manmohan Singh, the then finance minister, in

his famous 1991 speech in parliament that liberalised the Indian economy)

ANNEXURE-I

FAQs

1. Can I use Akruti or ShreeLipi with InDesign?

A: Yes, you can. But I strongly recommend against mixing Akruti/ShreeLipi with Unicode within one document.

2. You said Kalinga can replace Akruti Sarala and Shree 601, but there's something I don't like about the font (such as square-ish colon, semicolon and quote marks). Can you modify Kalinga?

A: Kalinga is copyrighted property of Microsoft corporation. But Lekhani project is working on developing a font in looks similar to Sarala/601. Until then, as a temporary measure. I've made a script that will change these symbols in Kalinga to another font of your choice with rounded forms such as Times New Roman. Mail me to get it.

3. Will the converters ever achieve 100% accuracy?

A: Akruti converter probably will, it currently has 99% accuracy. Modular has (probably) deliberately made it so hard for converting their encoding. ShreeLipi converter is currently at 95% accuracy and will probably stay there.

4. ShreeLipi 7.0 gives two Ex fonts which they call Unicode font. Can I use them like other Unicode fonts?

A: The Ex fonts are just like a shell company that is used for tax evasion. They exist and follow rules only on pen and paper (i.e- ନାମକୁ ମାତ୍ର). These are technically identified as Unicode but don't follow the rules set by Unicode Consortium.

Unicode system has space for 1,114,112 individual letters. They are divided into 17 planes or sections each with 65,536 letters (the maximum one OpenType font can contain). Out of these, BMP (Basic Multilingual Plane) or Plane-0 supports the alphabets of most of the languages except CJK (Chinese-Japanese-Korean).

Plane 15-16 are called PUA (Private Use Area). These are intended for people or companies to make unique symbols, logos or characters for their own use and map them to Unicode. These are not portable (similar to Akruti/ShreeLipi).

Odia characters should be present in the Oriya block (0B00–0B7F) of BMP. Instead Ex fonts map Odia characters to PUA codepoints. For example କ should be mapped to 0B15 according to standards but Ex fonts map it to E813.

Modular is a commercial company so obviously they don't want their customers moving to an open standard like Unicode. That will kill their business model. But they have no right to mislead customers in this manner either.

5. Are you really a doctor? (Yes, I get asked this question very often)

A: I'm a medical student. But I'm a Jack of all trades kind of person. I do creative writing (you're reading my manual, it should be obvious), programming, Web designing, font designing, graphic designing, photography etc besides medicine.

ANNEXURE - II

Technical Documentation

Unless you know how to read and write programs you need not read this section

Technical documentation for the tools I have made will be placed here when I get the time and inclination to write them. Besides helping a technically capable person understand my code it will also serve for my own future reference. That is mainly because I was too lazy to comment my code.

For the moment be content with this quote from Sherlock Holmes

> "Sherlock Holmes' quick eye took in my occupation, and he shook his head with a smile as he noticed my questioning glances. "Beyond the obvious facts that he has at some time done manual labour, that he takes snuff, that he is a Freemason, that he has been in China, and that he has done a considerable amount of writing lately, I can deduce nothing else."
> Mr. Jabez Wilson started up in his chair, with his forefinger upon the paper, but his eyes upon my companion.
> "How, in the name of good fortune, did you know all that, Mr. Holmes?" he asked. "How did you know, for example, that I did manual labour. It's as true as gospel, for I began as a ship's carpenter."
> "Your hands, my dear sir. Your right hand is quite a size larger than your left. You have worked with it, and the muscles are more developed."
> "Well, the snuff, then, and the Freemasonry?"
> "I won't insult your intelligence by telling you how I read that, especially as, rather against the strict rules of your order, you use an arc and compass breastpin."
> "Ah, of course, I forgot that. But the writing?"
> "What else can be indicated by that right cuff so very shiny for five inches,

> and the left one with the smooth patch near the elbow where you rest it upon the desk.”
>
> Mr. Jabez Wilson laughed heavily. “Well, I never!” said he. “I thought at first that you had done something clever, but I see that there was nothing in it after all.”
>
> “I begin to think, Watson,” said Holmes, “that I make a mistake in explaining. ‘Omne ignotum pro magnifico,’ you know, and my poor little reputation, such as it is, will suffer shipwreck if I am so candid.” ”

—Sir Arthur Conan Doyle (The Red Headed League)

* *Omne ignotum pro magnifico* = “All things unknown seem grand.” (and once explained, appear coomonplace.)

This page intentionally left blank