

Standards & Guidelines

For

**Localization & Language
Technology Standards**

Government of India
Ministry of Communications and Information Technology
Department of Information Technology
National Informatics Centre

Executive Summary

The success of National e-Governance Programme (NeGP) depends largely on the information dissemination and on the user-friendliness of its applications. This can be made possible by having the citizen and businesses interact with the government in their own languages. Therefore, effective localization of applications that address users' understanding and conventions is the 'last mile' of NeGP solutions delivery.

In order to under the issues related to localization of content and services under NeGP, the Localisation and Language Technology Standards Working Group held a series of consultation across the country with linguists, users, implementers and industry. Some of the key findings of these consultations are as under:

Findings:

Varied Support for 15 out of the 22 Official Indian Languages exists presently. The fifteen languages are : Assamese, Bengali, Gujarati, Hindi, Kannada, Konkani, Malayalam, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Tamil, Telugu and Urdu. The remaining seven languages, namely, Kashmiri, Maithili, Manipuri, Sindhi, Bodo, Dogri and Santhali are not yet supported in all platforms.

Though there are some gaps, it will be possible to create e-Governance applications in the supported languages for the relevant functions, using existing Indian and global standards. After implementation, interoperability is a pre-requisite for the applications therefore Global Standard like W3C has been considered important because of its openness and vendor support.

Issues:

The existing standards and resources for Indian Language computing are not complete in all respects. The gaps that exist have been identified as follows:

1. Operating System level support
2. Keyboard layouts and character formations (of conjunct characters).
3. Terminologies for Indian languages (both technical and non-technical).
4. Unicode points for some Indian scripts (such as Santhali and Kashmiri).
5. Lexical resources such as transliteration reference for Indian languages.

We recommend that the relevance of these be reviewed and action taken (i.e. the specifications/ resources be completed) as soon as possible.

Recommendations:

We recommend that:

1. All citizens-centric e-Governance applications that are available on the internet, implement functionalities equally in English and in the relevant official Indian Language.
2. Based on this, all NeGP applications be designed for localization from the beginning.
3. Technology providers and solutions implementers specify their level of support for the relevant languages and OS support issues for all official Indian languages may be resolved in a time bound manner especially for those 7 languages for which no OS level support is currently available.
4. Unicode may be notified as the standard character encoding scheme in place of ISCII and Migrating existing ISCII based data into UNICODE must be given priority.
5. Output of any user specific keyboard layout must conform to UNICODE current version.
6. Only Open Type Fonts may be used for NeGP applications/ solutions.
7. A small group of experts shall be constituted for each of the 22 Official Languages which will make a thorough study of the current status of all aspects of
 - a. Technology support (including character encoding schemes, input methods, OS and browser support, inter-conversion between different formats such as Pdf and PostScript, search and processing etc.) for the concerned language script, identify gap areas and suggest necessary action plan for bridging the gaps quickly.
 - b. Lexical resources (including corpora, dictionaries, morphological analyzers, thesauri and wordnets, spell checkers etc.) for the concerned language/script, identify gap areas and suggest necessary action plan for bridging the gaps quickly.
8. A list of all existing applications using any degree of localisation may be compiled .A pilot study in localization of selected G2C e-Governance applications using the recommended standards shall be carried out within six months. This will help formulate guidelines and priorities for further research and development in relevant areas.
9. A database of relevant named entities such as place names, names of companies, organizations and person's names be compiled on a priority basis and transliterated into all the 22 Official Languages/Scripts. The transliteration may be done from Indian Language to English and not vice-versa.
10. Training Programmes may be conducted to train different classes of users on the use of local language technologies.
11. Existing Speech-to-text and Text-to-speech prototypes/products may be tested for their usability in addressing localisation issues.

**Localization & Language Technology Standards
Standards/Guidelines for National e-Governance Plan applications**

Application component	Present Status/Standard	Need of Standardization for NeGP	Recommendations & Roadmap
Encoding	<ul style="list-style-type: none"> ISCII, Unicode and Proprietary codes are in use. 	Unicode	Unicode characters are almost complete to suffice the respective language requirements.
Keyboard Layouts	INSCRIPT keyboard layout. Various adhoc keyboards are also used such as typewriters, phonetic and glyph-based.	Existing INSCRIPT standard should be upgraded to include new Unicode code points for all Indian Scripts.	Output of any user specific keyboard layout must conform to Unicode current version.
Font	No Fonts are standardized so far.	Cross platform Open Type Fonts must be used in National e-Governance Plan applications.	Only Open Type Fonts may be used for e-Government applications
Content Creation - Browser Support	So far no standardized Content Creators are available.	W3C Standard	<ul style="list-style-type: none"> Adoption of W3C specifications. All citizen interfaces should carry the W3C certified logo. All the content created should conform to the W3C specifications as listed here: <p>http://www.w3.org/QA/Tools/#validators</p> <p>For further understanding, see: http://www.w3.org/QA/2003/03/web-kit</p>
Search/ Cross Lingual Search/ Transliteration	There is no existing standard so far.	There is no existing standard for Search/ Transliteration. If database/ contents are in	Transliteration tools in the NeGP are extremely important since it is presumed that a large part of the information on the identity of users will be in their local languages. Therefore, at the very least this information

		<p>Unicode/W3C format, then there is no problem in Search.</p> <p>Unresolved.</p>	<p>will have to be transliterated into the scripts of users of this information in different states.</p> <p><i>A database of relevant named entities such as place names, names of companies, organizations and person's names be compiled on a priority basis and transliterated into all the 22 Official Languages/Scripts.</i></p> <p>For example, a Smart-Card based driving-license issued by an Regional Transport Office may serve a restricted purpose if any information in one script cannot be read in different parts of the country. Though there are other examples, the criticality of a 'perfect' transliteration tool will depend on the criticality of local language data in a given NeGP application.</p> <p>The transliteration may be designed from Indian Language to English and not vice versa</p>
--	--	---	---

This section delineates the broad policy guidelines and recommendations for standards with regard to localization and language technology applications in India, especially in the context of e-Governance. Localization and Language Technology Standards are very vital for the success of e-Governance initiatives, especially for G2C and G2G sectors.

1 **Goal:** Whatever can be done with computers in English today, we should be able to do the same in Indian Languages/Scripts with equal ease.

2 **Language and Script Independence:** All systems (including relevant hardware, system and application software, services, applications, tools and technologies, documentation, help etc.) shall be fully compatible with all the 22 constitutionally recognized languages and corresponding scripts.

3 English will continue to be fully supported.

4 Functions in NeGP web applications should be such that they can be used from existing/legacy systems that are still in use today. (Existing/legacy systems that do not fully conform to the accepted standards shall gradually be upgraded to become fully compliant in a phased manner.)

5 All systems shall be designed in keeping the national (BIS) and International (ISO) standards as also de-facto standards such as UNICODE and W3C consortium standards.

6. Given the facts that a) there are a large number of languages spoken in India some of which do not have a script of their own, b) the large population with relatively low levels of education, c) complexity of using conventional computer systems, and d) low levels of fluency in English, current keyboard-mouse-screen based systems will be too difficult for most people to use. The future is in speech based interfaces. Importance shall therefore be attached to development of suitable speech based and multi-modal communication systems, especially for G2C applications.

Character Encoding

All documents must be encoded in a standard character encoding format.

ISCII is currently the only character encoding standard approved by BIS (IS 13194:1991). UNICODE is an existing ISO standard. Unicode is based on ISCII and not too different from it.

UNICODE shall be the character encoding standard. Suitable action may be taken to legally permit the use of UNICODE in India. Existing ISCII data will have to be migrated to UNICODE in a phased manner.

- ISCII recognizes that languages and scripts are fundamentally different from one another and the mapping between language and script neither needs to be nor actually is one-to-one. ISCII is for scripts, not languages.

- ISCII recognizes that linguistics units such as words and sentences have a) meanings, b) pronunciation, and c) orthographic representations associated with them and of these three, pronunciation is the second most important aspect (after meaning, which cannot be handled by computers today). Languages need not have a script, orthography is secondary. Indian scripts are based on pronunciation -based on the fundamental sound units called phonemes in a language, not on written shapes. Sounds are more universal than written shapes. Written shapes also evolve and change with time but the sound system of a language does not change significantly. ISCII is based on this fundamental understanding of the true nature of Indian languages and scripts. The codes in ISCII represent sounds, not shapes. This is a very wise decision and thinking purely in terms of written shapes would be inappropriate in the Indian context.

- ISCII recognizes that the sound system of Indian languages is largely common. Indian scripts are largely common in terms of the sound system, although the written shapes vary widely across scripts. ISCII therefore proposes a hyper-script, a script that is a small superset of all the units required for all the 10 brahmi-based scripts. This is also a very wise decision that promotes unity and integration as against local, narrow, sectarian, separatist proposals.

- ISCII recognizes that Indian languages are unique in having a script grammar -a grammar at the level of script, not language. ISCII standard (IS 13194: 1991) specifies the script grammar. The script grammar allows all valid combinations and disallows all invalid combinations. Without this, impossible and meaningless shapes will be allowed.

- ISCII is silent on fonts. There are no ISCII-aware fonts, fonts that can automatically accept and handle ISCII. There are no font encoding standards for Indian scripts -font developers are free to select glyphs and glyph variations and position them wherever they want in the font table.

- ISCII is silent on character to font mapping, a necessary step for rendering Indian scripts for the purposes of display and printing. This mapping can be and often is very complex. A provably complete, 100% consistent, simple, elegant and computationally efficient mapping scheme is lacking, forcing software developers and vendors to resort to ad-hoc, inefficient and erroneous implementations.

- ISCII was never registered and no Operating System and no Browser automatically supports ISCII. You can buy any computer from any vendor anywhere in India and expect to be able to type in a letter, save it, print it and do all such basic operations in English without having to buy or install any specialized hardware or software or font. The case is not so with Indian scripts. Localization and specialized solutions are explicitly called for.

- UNICODE is based on ISCII and hence inherits many of these properties except as brought out below.

- UNICODE confuses between language and script. It has provided for a common code space for Bengali (Bangla) and Assamese, just as in the case of Chinese-Japanese-Korean, but at the same time tends to think in terms of languages rather than scripts. Why not a common code space for all Indian scripts as in ISCII? Why not separate code spaces for Assamese and Bengali? The decisions seem to be based on visible shapes of orthographic renderings rather than the fundamental sound system of the languages in question.

- UNICODE is silent on script grammar, an essential and unique feature of Indian scripts.

- UNICODE does not provide a sound system for character-to-font mapping. Instead it expects the fonts to somehow handle this complex task. UNICODE fonts are expected to include context based glyph selection and composition rules. This is not very practicable, especially for complex scripts. UNICODE based systems today face a whole gamut of ugly problems relating to glyph selection and composition.

- UNICODE is supported/can be expected to be supported by all major operating systems, browsers, editors, word processors and other applications and tools. It will be possible to use Indian languages and scripts in the UNICODE environment without need for any specialized hardware or software. Indian language support will be naturally and automatically available everywhere.

- UNICODE is widely recognized all over the world and switching over to UNICODE will enable to get global recognition and interface seamlessly with the rest of the world.

In summary, ISCII is difficult to use today because there is no font-encoding standard and there is no standard way of mapping from character to fonts. We have to shift over to UNICODE.

Existing ISCII based data needs to be ported to UNICODE. ISCII-to-UNICODE converters do exist but the conversion may not be 100% accurate. Development of high performance ISCII-UNICODE converters should be taken up on a priority basis.

White Paper on Indic Fonts for National eGovernance Plan

Note: This is a brief overview of the considerations for choosing a font for Indic computing in NeGP. It does not cover all aspects of font technology.

Fonts

The elements of the script of a language are characters. The shapes of these characters are called 'glyphs'. The number of glyphs in a script is the total number of shapes that those characters can take. They are equal to or more than the characters in the script.

A font is a resource used to print characters in a script on a computer. It contains:

- 1
1. The glyphs that comprise the characters in a script.
- 1
2. Some tables, as required, that contain information about how individual characters combine to form final glyphs.
3. (For Indic computing, the relevant font-technology is called [Open Type](#) . The underlying character-set of a font is based on its [Unicode representation](#) .)

A layout engine, like [Pango](#) or [Uniscribe](#) , interprets the information in a font to represent the script on the output device. Layout Engines and Fonts work together for the final result. A shortcoming-in- or incompatibility-between either of the two software prevents the correct formation of glyphs in a script.

To be 'complete', a font must:

- 1
1. Ensure that all the glyphs in the script are present and formed by the correct combination of the relevant base characters.
- 1
2. Contain the relevant information in the tables to form the final glyphs.
3. (A font is 'complete' only in the context of the intended application. e.g. A Devanagari font for Hindi for word-processing will not be the same as one for Marathi, Sanskrit or Vedic Sanskrit.)

The associated Layout Engine must:

- 1
1. Be able to interpret the information in a font to form the glyphs.

1

2. In general, comply with the "[Open Type](#) " specifications.

Fonts and Layout Engines should be tested together.

In any output operation, the objective is to represent the glyphs that are commonly used in the script. Therefore when one adopts or standardizes a font, one presumes that it has been tested on the relevant layout engines. This is not always to. Variations are usually caused by differential implementation of the Open Type Specifications by vendors.

Following are the recommendations to bypass the problems and to meet the objectives:

1

1. Customers should only adopt fonts that have been tested on different platforms for the relevant scripts.

1

2. Proprietary fonts should not be adopted unless the vendor assures interoperability.
3. Vendors should ensure that the fonts provide the complete glyph-set and associated tables for the end-result to be met.

White Paper on Operating System Support:

Overview:

The Operating Systems (OS) is the fundamental software program that uses a computer's resources to deliver useful capabilities to users. In this role this is the place where the resources for local language computing are present. Hence, the enablement of Indic Languages in OS is central to the availability of interoperable Localized applications.

Elements of Operating System Support:

Level 1: Data handling.

Level 2: Enablement of the UI

Level 3: Advanced Support.

Level 1: Data Handling (for Indic).

This is the most basic level of language support whereby the character-repertoire of the script of a language is enabled for support. The character set uses Unicode specifications.

Thereafter the OS is enhanced with the cultural elements of the languages. These are:

1

i. Character Display/Output: How the script appears in display or when printed. This ensures that the when the characters in the script are combined, they form the expected output. It's important to know that the output mechanism is dependent on the logic used to specify the character set. This is relevant for Complex scripts like Indic because the characters combine to form compound characters. For Indic, the character formation logic is based on the ISCII model.

1

ii. Character Input Method: i.e. How the characters are entered into computing machines. This is related to the display method because the display of characters is related to the linguistic model for their formation. The default input method being via a keyboard. For Indic scripts, INSCRIPT is the default method. It's important to note that INSCRIPT has not been updated to include the additions to the Unicode Code Spaces for Indic. This is a serious problem because if vendors add these to locations of their own choosing, then the standard will break.

iii .Locale: Description of culturally-required data like currency, date-time information, sorting-order, calendars, etc..

The output of this activity is a "functionally complete" computing environment for users of an individual language.

The Unicode Common Locale Data Repository (<http://www.unicode.org/cldr>) is the repository for locale data. Data for Indic Languages may be viewed, enhanced/ corrected and created for the cases in

which they do not exist.

Microsoft's Locales information may be found here:

<http://www.microsoft.com/globaldev/DrIntl/faqs/Locales.msp>

N.B.: 'Functionally complete' should be considered a relative term. It is possible that some aspects of an OS' resources may not be localizable or localized because of their impact on the network it is connected to. For example: It may be reasonable to assign a computer a name in a local language. But this would impact the systems that would manage them and other capabilities like provisioning, These logical (in)consistencies should be evaluated before a policy is adopted.

Level 2: Enablement of the UI:

The next step in enabling the use of a local language version of the OS is to provide a multilingual UX (user experience). This enables users who are monolingual to use (User Interface -UI) and to seek help (User Assistance -UA) about the computer in a language they are conversant with. While depending on the manner in which the operating system is architected this process of enabling the user experience largely involves a combination of terminology definition, translation and attachment of skins to the operating system.

Functional completeness must be augmented with User Interface translation to make a computing device usable by someone who doesn't know / is not proficient in English.

Translation may be broken up into further levels:

- 1 i. User Information: This is the information that the user encounters on a daily basis.
- ii. Program Information: This is information that is integrated into the programs, such as help files, etc.. Translation of this is dependent on the nature of the product as well.
- iii. Complete Translation: Translation of *everything* that is relevant to a user's environment. This is also dependent on the nature of a product.

It is critical that Operating Systems present an Indic UX to the users, as a majority of the literate population in rural India is monolingual in non-english languages. This will be important for disintermediation of the use of applications. i.e. to enable users to be able to use solutions on their own rather than depend on assistance. However, this may not be applicable for OS that function as servers. This is because the developers and administrators of applications will all be proficient in English. In some of these cases an Indic UI may hinder rather than aid an administrator's work.

Level 3: Advanced Capabilities.

The above two activities prepare an OS to be usable in a local language. These may be enhanced by improving the usability of the OS. These may be termed as 'Accessibility Enhancements', though it is a term that does not strictly apply to this domain.

Certain usage scenarios require the users to interact with the computer purely based on voice. In such a situation operating system. These are incremental features that

1

1) Handwriting recognition

1

2) Voice recognition

Handwriting/Voice support for Indic does not exist but is in process of being developed.

Accessibility enhancements like Voice and Handwriting Recognition, etc. provide more intuitive input/output capabilities to users and hence allow applications that make use of those resources to reach more people.

Most OS like Windows, Linux, AIX etc. have the capability for Accessibility Tools. Independent vendors also have solutions that run on these OS.

Things to look into Operating Systems:

If Localization is important when applications are being built, following are the things to look for to ensure their interoperability across client-devices:

1. Unicode support for a language and script: Vendors should ensure that the implementation of Unicode in their Operating Systems support the entire character set of the *script*: It should be re-iterated that some of these need monitoring and action from CDAC/TDIL. Though industry aids these decisions, it cannot be responsible for these.

2. Claimed level of user interface translation: This decision depends on the policy of the government to provide local language capabilities in the end solution. Translating a UI without there being a corresponding need in the solution will be wasteful and unreasonable.

3. Claimed level of support for W3C standards in web-browsers on those platforms: Since W3C has been recommended as a standard to measure inter-operability of applications, the browsers that applications are tested on, must be W3C compliant, i.e. capable of supporting the W3C specifications.

4. The ability of data created in one OS to be used in another: Under no circumstances should any application create unencrypted data that cannot be viewed in another platform OS. This would be subject to the above criteria. In case such data exists there should be tools for conversion or ISV's be enabled to develop these.

Recommendations:

The achievable local-language support in an application is subject to:

1

1. The goal of the policy.

1

2. The completeness of the decisions to permit the implementation of the policy.

The Operating System is the bedrock on which the rest of the application runs and is used. It is critical that the decisions be made with full-knowledge of what they will permit or not.

These decisions will affect the longevity and long-term stability of an application. We recommend that the decisions are made with full understanding of the repercussions.

-

White Paper on Transliteration for National e-Governance Plan**Note:**

This is a White Paper which elaborate on the role of transliteration in the National e-Governance Program. It is not an exhaustive view on transliteration. A detailed paper may be prepared if required.

Definition:

Transliteration is the conversion of text from one script to another.

This is different from translation which is the conversion of information from one language to another.

Function:

Since transliteration is not translation its usefulness is limited to situations when the user of the transliterated information :

Case 1. Understands the language and can read one (or both) scripts.

Case 2. Does not understand the language but can use the information for the intended purpose.

<i>User</i>	<i>Scenario</i>	<i>Usage</i>
Understands the language and can read one (or both) scripts.	Conversion of unformatted/ unstructured information, such as news, from one script to another.	When a language is written in different script, this is used to make information in one script available in the other script.
Does not understand the language but can use the information for the intended purpose.	Conversion of selected and relevant information from the source script to a script readable by the user.	Because it is not translated, only the information that can be "useful" is converted. Example: Identity and address information in electoral rolls.

The characters which comprise different scripts may not correspond phonetically. Hence it may not always be possible to represent the sounds in a language that are in the native script, in a different script. When transliteration is used to ensure phonetic correspondence between different scripts, it is called Transcription. Machine-based Transcription is not always reversible, i.e. it is not always possible to convert Transcribed text back to it's original form.

Application area:

Transliteration tools in the NeGP are extremely important since it is presumed that a large part of the information on the identity of users will be in their local languages. Therefore, at the very least this information will have to be transliterated into the scripts of users of this information in different states.

For example, a Smart-Card based driving-license issued by an Regional Transport Office may serve a restricted purpose if any information in one script cannot be read in different parts of the country. Though there are other examples, the criticality of a 'perfect' transliteration tool will depend on the criticality of local language data in a given NeGP application.

Usage in NeGP:

If the NeGP includes robust capabilities to offer transactions to users from the Common Services Centres, then identity management of users will be a major component. Correspondingly, these identities must have a local language option. It is also likely that some of these identities may be usable in areas where the the local language is different. (An example of this is the Reservation/Passenger List printed by the Indian Railways.)

Transliteration tools will be required to resolve the script differences.

The size of the transliteration system depends on:

- 1 Domain and Type of application.
- 2 Number of users.
- 3 Reach of the application (presuming that some applications may not be delivered everywhere).

It is possible that transliteration be used to serve a limited purpose in a limited application domain. It is be desirable that each domain contain data in all the relevant scripts, to enable the scope of the capability to be replicated across languages.

Transliteration of unformatted/ unstructured data is unlikely to be of any serious use at this point of time. It's possible that this may change in the future.

Recommendations:

The nature of transliteration is such that there can be no universal algorithm to address Indic scripts. Indic scripts are also such that though they have common linguistic and script roots, they have evolved to create their own characteristics so that one-to-one character-matches are not always possible.

For this reason, the best approach for maintaining transliterability of Indic data-elements is to also create the matches for those elements at the time of creation. The resulting data/ dictionary should be cloned across the application's infrastructure to ensure consistency. Alternatively, a transliteration web-service(s) may be defined that could be used by several applications.

Clearly, project managers for various NeGP applications must identify the role of transliteration in their solution and take the decisions. These are the recommendations from a policy perspective:

1 Use a "standard" transliteration algorithm and enhance deviations with a dictionary.

2 Ensure that the data/dictionary is open, to enable it to be enhanced and re-used across different NeGP applications.

3 If a vendor claims to have fool-proof transliteration, then the service levels must be validated accordingly.