

# 22

## Software Localisation Process and Issues

**N. Anbarasan**

Chief Executive Officer, APPLESOFT  
39, 1st Cross, 1st Main, Shiva Nagar, West of Chord Road  
Bangalore - 560 010  
<E-mail: arasan@bgl.vsnl.net.in, aplesoft@vsnl.com>

---

### Abstract

Language interface to software is more than just a requirement. However, as the localization process is tightly coupled with the underlying operating system, the language interface has been evading. As the operating system is maturing to handle multiple languages, the approach to language interface technology needs a revisit.

As of today Localisation of MS Windows Operating System has not been successfully attempted by anyone than Microsoft. With the achievement of Localisation product 'Language Interface Pack' developed by Microsoft, it is now possible to localize MS Windows XP Professional into 33 languages of the World. The other unfortunate languages including Indian Languages have to wait for their turn to experience the successful use at the OS level.

This author has been attempting localization of operating system and application software since 1989. As a result of today's multilingual technology, it has been made possible to handle language interfaces of operating system and application software. The technology thus developed is an evolution in the area of localization technology and is capable of handling user interface of Microsoft Windows operating system and application software running on it.

This paper tries to explore the different localization methodologies being tried in the industry and present a case study on the issues involved.

### Localizable software

The kind of technology and tools available today makes the software development process specific to the requirement. As the interface is directly dependant on the field of requirement of software, the types of interface elements also varies. Even though, a software may seemingly contain string interfaces but actually would contain images. As such the tools and in turn the technology used for developing software poses a bottleneck for localization.

For a successful localization of product, the product should have been developed using standard convention like Resource files. Any product developed using non-standard conventions could not be localized.

In general, the software developed with menus, dialog boxes, message boxes, status bar texts and tool tips are localizable without the source code. This type of software includes, operating

system, application software, tools etc, whereas the multimedia-based software can be localized only by the developer and involves almost equivalent development cycle as that of the original. As the multimedia-based software also contain graphics, depicting the cultural identity, the localization process involves change of graphics related to culture, lifestyle etc.

### **Language Engineering**

Language Engineering is the process of codifying the human knowledge of a language like alphabetic system, script, grammar etc to enable the software to utilize the knowledge. As a result of language engineering, the software would be able to recognize, interpret, understand and generate human language. The resultant form of language engineering would be encoding of language, script, grammatical components like phonemes, morphemes, sandhi rules, dictionaries etc. These components are extensively used in rendering script and in natural language processing applications like Machine-aided Translation, Speech Recognition, Speech Synthesis etc.

Due to contrasting encoding methods and difficulties in adopting the proven traditional grammar, Indian languages are not able to achieve any break through. Indian languages suffer due to lack of any standards based on the grammatical rules. In recent times, ISCII, the only script standard for Indian languages is increasingly adopted by software developers.

The language specific information are stored as locales in the Operating System which include Sort order, Input method, Data & Time, Number and Currency formats. MS Windows XP supports 135 locales.

### **Localizable Resources**

The localizable resources generally include Bitmaps, Icons, Sounds, Menus, Dialogs, Number and Currency formats, Date and time formats, Messages, Prompts, Status bar, Tooltips, Input methods and Help files. Attempts are being made to localize the text based user interfaces by translating the text. There are some translation tools which supports machine translation and dictionary for look up etc.

### **Localization Process**

The process of adopting and customizing a product for a specific market is localization. Localization involves translating user interface elements like Icons, Menus, Messages, etc. But, normally only the user interface Texts like, Menus, Dialogs, Messages, Prompts are Translated.

### **Localization Methods**

Localisation has been tried by varying methods. However, every method has been developed for certain requirements. Some of the popular localization methods are:

#### **Source code localization**

The localization process started with localization of source code and it was carried out by the original developer. The steps involved in the localization on process with source code is :

1. Extracting user interface strings: In this task all the user interface strings like Menus, Dialogs, Messages, Status bar text, Tooltips etc are extracted from the source files.
2. Adapting language specific conventions: The user interface strings are translated and number and Date & Time formats are applied. Appropriate character set, Code pages, Font names are also adapted and applied as per the requirement. String buffers are adjusted to store the translated strings. This also means altering and

adjusting of dialogs to accommodate the translated strings. If enough care is not taken at this point truncation of strings would result in incomplete translated strings being displayed.

3. Compile and testing: After adjusting the language specific interfaces, the source code is recompiled and is ready for testing. On successful testing, the product is ready for the market.

This method is still continuing in the open source community. The established software organizations like Microsoft also followed this method for the earlier products.

**Executable Localization:** As the runtime binaries are matured and PC penetration increased, the localization need for various applications have also increased. And also there is a compelling reason to localize the product and release in local market simultaneously. To meet these requirements, the software developed should take care of the international requirement at design and development of English version itself. Thus developed product will be sent to localization team, the localization team in turn extract and translate the user interfaces using some localization tools, normally with visual testing capabilities. The translation and testing goes on hand in hand. This localization process normally takes less time compare to earlier source code localization. The product is distributed with translated resources.

**Runtime Localization:** The recent development in localization is to localize the software during runtime. In this method also the localization was carried out with running process in the memory. But the resources are not embedded into binaries but distributed separately.

### Localization Tools

There are varieties of localization tools available with varying capability and features. The simplest one would extract the user interface strings from the specified binaries and give out a text file for translation of user interfaces. The advanced localization tool is available with editing facility for resources and machine translation capability. This type of localization tools are available for only a few languages.

### Localisation efforts at APPLESOFT

APPLESOFT was started with a singular motto of developing innovative language technologies. During 1990, a project to localize MS-DOS was started to enable all text based applications to support Indian Languages and the same was successfully completed during 1993. The product was demonstrated at Vedanta South, the power center of Karnataka during the inaugural day celebrations of Kannada celebrity year.

When MS Windows 3.1 became popular, the localization project started on for MS Windows and the same was demonstrated at TDIL (Technology Development for Indian Languages) meet held at New Delhi during 1998.

### Janine

When Microsoft announced its new year Operating System - MS Windows 95, our localization effort continued further to support MS Windows 95. As a outcome of Tamilnet99 Tamilnadu Govt. has setup TSDF. Our localization project got financial support from TSDF.

### Features of Janine

The first ever run time localizer, Janine has been developed to meet the requirement of End-user, Translator and software developer. Some of the features of Janine are:

1. Bundled with utilities to enable the system with bi-lingual font, to capture user interfaces and an editor to help the translator.
2. Bi-lingual capability with English and Tamil. The user can opt the language from the menu.
3. Can be used to localize any application developed using standard windows techniques and resources.

### Shortcomings of Janine

As the Indian languages do not have any International Standard implemented on MS Windows 95/98. Janani has been developed using bi-lingual fonts. Hence Janani having the following shortcomings :

1. Font based technology. Hence supports only those scripts which can be rendered using a bi-lingual font.
2. As the translated strings are normally lengthier than the original English string, the display of strings are limited to the space available for English strings. Hence the strings which are lengthier than the display area are cut-off resulting in incomplete strings.

### Localiser

In order to overcome the shortcomings of Janani, another project was undertaken and code named localizer. Localiser is the first run-time localization tool developed to localize the user interface strings of running applications on MS Windows. Localiser changes the user interface strings like menus, dialogs, status bar texts, tool tips etc.

Screen capture of MS WordPad is shown in figure 1.

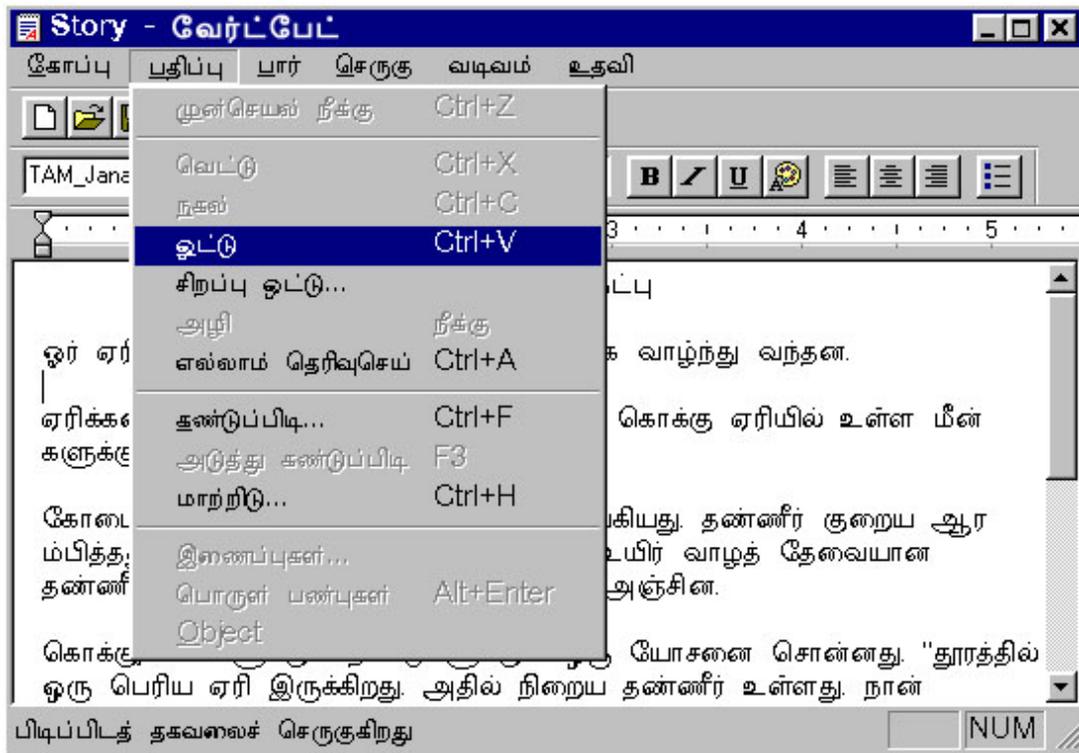


Figure 1

Localiser has been tried with standard applications like Internet Explorer and Control panel applications. Screen capture of IE is shown in figure 2.

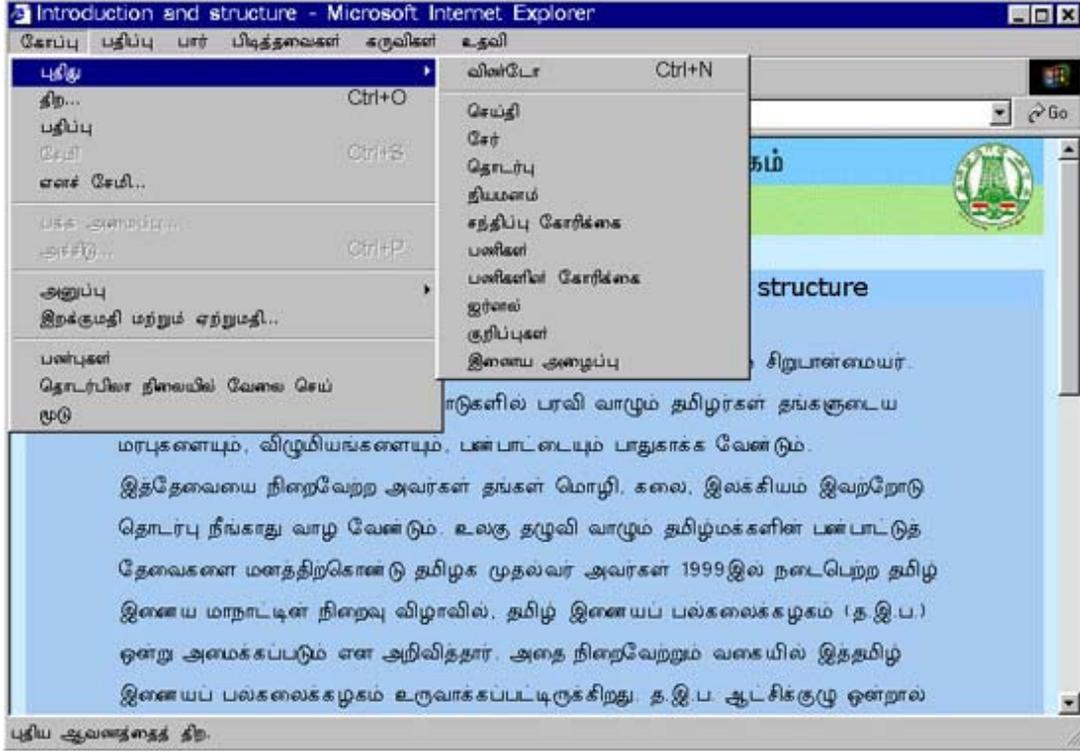


Figure 2

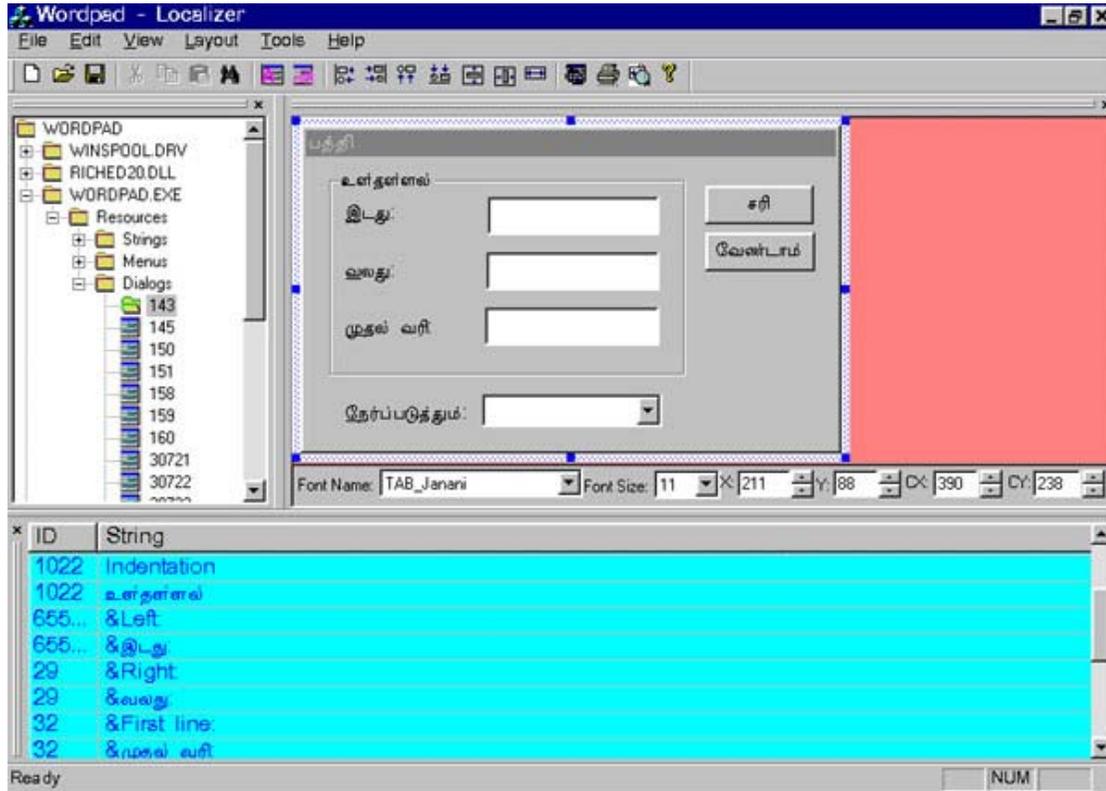
### Features of localizer

The localizer has been developed with Multi-lingual capability and to support wider operating systems such as MS windows 95, 98, ME, NT, 2000, XP. Presently, the localizer is having a range of features to meet the requirements of the end user, developer and translator with visual tools. The translator tool is shown in figure 3. Some of the features of localizer are:

1. Applications running on MS Windows operating systems are localized during run-time. No need for any source code for the application to be localized.
2. Varieties of visual tools for developers and translators.
3. The visual tools come with built-in resource editors to handle menus, dialogs.
4. Resources preview shows menus, dialogs
5. Handles wide range of file formats and help contents.
6. Unicode enabled.
7. Support wide range of Microsoft operating system and applications running on it.

## Issues in Localization

In spite of all the technical advancements in localization, the success of localization really lies in the application to be localized. Unless the application is developed to meet the international requirement, it is not possible to localize the binary version of the application. There are various issues associated with the software development which would hamper the localization. Such issues can be broadly grouped as string constants, Character range, Buffer size, Data type.



**String constants:** There are different types of hard coded string constants which includes Names, File names and File paths.

If the application uses name for string comparison then the application can be localized only by the original developer by modifying the source code. Similarly, use of hard coded file-names and File paths poses problem for localization of help files, data files used by the application.

**Character constants and character range:** When the application requires any confirmation, it would have been implemented using character constants. For example 'Y' for Yes, 'N' for No etc. This hard coded character constants can be localized only by the modifying the source code.

Similarly, character range is also used to validate the data entered by the application software user.

**Buffer size:** The translated resources are normally bigger or lengthier than the original English resources. In case, if the buffer used to store the English resource is of fixed size then the localized version might result in incomplete strings.

**Display area:** Similarly, the strings displayed in a dialog box would also be truncated due to the pre-fixed position of various controls and their sizes.

**Data type:** Data types play a crucial role in input data types. If the application is designed to hold 7-bit data, which is sufficient for English text, is not capable of holding any other language. It is due to the fact that non-English language script normally needs more than 7-bit.

**Conclusion:** It is observed that there are different methods of localization being practiced. However, due to the limitations in the tools used for localization and applications limitation to cope with localization, it stands as a distant dream for 100% localization. That being the state, adapt-ing and achieving localization for an Indian Language is an indelible mark of innovation to offer "knowledge for all".