Tibetan For Windows - Software Development and Future Speculations

Marvin Moser, Tibetan for Windows & Lucent Technologies, USA

Introduction

This paper presents the basic functions of the Tibetan for Windows software program, with a comparison to a related program, Tibetan on the Macintosh. Some of the processing algorithms and technology are described, along with how the program can be modified to accommodate other Tibetan fonts. The paper concludes with a review of current trends in Tibetan computing and some speculations on areas for future development.

What is Tibetan For Windows?

Tibetan For Windows is a Windows program designed to make entry and editing of Tibetan as easy as possible. It features a "what you see is what you get" approach, displaying Tibetan characters as one types. There are three transliteration schemes supported: Wylie, phonetic and Tibetan typewriter. The program works with the popular text editors Microsoft Word for Windows and WordPerfect for Windows. Tibetan text can easily be cut and pasted between many different Windows programs.

Tibetan files can be exchanged to and from a PC and any Macintosh computer equipped with the Robillard LTibetan font, allowing both computers to work on the same project. Printing can be done on any of the graphics printers supported by Windows, including dot matrix, laser and ink jet. (See the appendix for font samples.) A free utility program is also included, the Tibetan File Converter, which runs under either DOS or Windows and converts between any combination of Wylie, ACIP and the Robillard font. Phonetic transliteration can also be generated as output from any of the above formats.

There is a strong connection between Tibetan For Windows (TFW) and the Tibetan on the Macintosh (TOTM) program by Pierre Robillard. Mr. Robillard has kindly provided both his font and source code for conversion to the Windows environment. I also licensed Wylie parsing code from Chet Wood. The upcoming version 2.0 of TFW will incorporate new ACIP Wylie parsing code from Mr. Robillard. The addition of any missing or seldom used Tibetan or Tibetanized stacks will then be very easy, requiring only the editing of table files using standard text editors. In addition, the entire TFW and TOTM source code is planned to be made generally available free of charge under the GNU licensing agreement, permitting full flexibility in modification. As character definitions are done primarily in the table files and in header files, modifying the source to allow new Tibetan fonts should be very straightforward.

Tibetan Word Processing Functions

There are three major steps involved in processing a user's typed input for Tibetan word processing. These steps are very general, and can be used for comparing many word processing packages. They are:

- Keyboard Mapping
- Parsing (optional)
- Character Display

Keyboard Mapping defines how individual keys on a keyboard are mapped to characters in a given language. The Tibetan For Windows (TFW) program allows three different predefined mappings: Wylie, Tibetan typewriter and phonetic.

The Wylie mode uses the standard QWERTY key layout to enter the characters of the Wylie transliteration method. Since Wylie transliteration is designed for Roman characters, users who already know how to type generally find this the fastest way to learn to enter Tibetan. In fact, users who know how to type and how the Wylie transliteration method works can usually sit down and begin entering Tibetan text with virtually no training. TOTM allows Wylie input using the Wylie Edit program written by Chet Wood. TOTM also has the Marpa editor (again by Pierre Robillard), which allows direct entry of the ACIP variant of Wylie transliteration.

The Tibetan typewriter mode assigns keys to Tibetan characters sequentially, starting with q representing the letter . This input mode is useful if the user has little or no previous typing experience, and is a native Tibetan speaker. Interestingly, this mode is not used in at least one large text input project with native speakers, as teaching them Wylie allowed them to also learn the QWERTY layout, a valuable skill for future employment. TOTM uses typewriter mode with the required MacKeymeleon keyboard customizer program.

The phonetic mode assigns keys to Tibetan based on the sound of the Tibetan characters. Thus, k would be . This is perhaps the least used input mode. TOTM does not currently support phonetic input.

Parsing is the identification of the role of a character in the context of its syllable. Because the vertical position and size of Tibetan letters depends on this context, parsing is necessary if automatic stacking of characters is to be performed. TFW uses parsing to automatically stack in all three input modes. This step is sometimes not present in other Tibetan software systems, requiring the user to press additional keys to indicate stacking position. In the TOTM system, WylieEdit and Marpa use parsing, while MacKeymeleon does not. As will be discussed later, TFW can be modified for other Tibetan fonts, enabling its parsing functions to be easily applied.

Character Display is how the Tibetan characters are presented on the screen. TFW comes bundled with the Robillard Tibetan font in both True Type and Postscript formats. These fonts can be used with any Windows program. However, due to the variation of key stroke handling by different Windows programs, the input mode and parsing functions are currently handled for only Word for Windows and WordPerfect for Windows. (Under Windows 3.1, the WordPad editor is also supported.) While many users work exclusively within these programs, other Windows programs, such as PageMaker, can use Tibetan files by either importing them or using cut and paste methods.

The Robillard Tibetan fonts are based on 8 bit ASCII character encoding. As such, files created using this font are portable between Macs and Windows machines. A Tibetan file conversion utility is provided with TFW, which can convert files in the Robillard font format into Wylie, ACIP or phonetics. Conversion of Wylie or ACIP files into Robillard format is also provided.

Wylie Conversion Algorithm

The conversion of Wylie transliteration into characters in a particular Tibetan font can be done in a number of ways, but the method used in the Marpa editor provides good flexibility by defining two phases: parsing and string lookup. Tibetan syllables are separated by special punctuation marks, allowing easy segmentation. Parsing within a syllable is done by scanning left to right, looking for the vowel marking. Thus, there are three sections within a syllable: the prefix, vowel and suffix. The Mac Marpa editor has lookup functions for each of these parts. The lookup tables for prefixes and suffixes are defined in external files read in at initialization. Tibetanized Sanskrit and partial stacks are also defined in additional files, and are processed similar to prefixes. Vowels and punctuation are defined in source code header files.

The lookup tables and headers primarily define the mapping from particular Wylie character sequences into Tibetan font character sequences. Thus, it should be possible to use the TFW keyboard driver with fonts other than Robillard by modifying the table and header files. In fact, if an ACIP converter for the new font already exists, one could use it to automatically generate the specific character sequences, and thus speed the creation of the new files. Thus, many Tibetan fonts that have been lacking a direct Wylie input method could easily have one.

Technology

Tibetan For Windows is written in two languages: C for the Windows keyboard driver code (including the hooks into the Windows API for catching key strokes directed at different programs), and C++ for Marpa Wylie parsing and table lookup. The Borland C++ 4.52 is being used in the current version 2.0 development. The conversion utility used Borland ObjectVision (replaced by C++ Builder for version 2.0) for constructing a windowing shell to specify input and output files.

TFW continues to evolve from its initial release in June 1992. Several point releases fixed font and program bugs, and allowed installation on Windows 95. The current work on version 2.0 will allow the Tibetanized Sanskrit input method (based on Marpa code). Compatibility with Windows 98 and Windows NT is also under investigation.

It is hoped that the availability of TFW and TOTM source code, free of charge under the GNU license, will spur porting of code to other platforms such as Linux, as well as further enhancements in the Mac and Windows environments. Anyone interested in such a project is encouraged to contact either myself or Pierre Robillard.

The Future of Tibetan Computing

Predicting the future is always a dangerous proposition, but I would like to make several observations, and then propose future directions that Tibetan computing in general might take.

The current development scene is very much influence by the very small market. There are very few developers, and of these, even fewer are engaged full time in development. One way to help this problem is through increased collaboration between developers. For example, the development of TFW has been greatly aided by contributions from Pierre Robillard and Chet Wood. Fostering the exchange between developers, such as at this PNC/EBTI conference and through email or list serve, can be very beneficial.

Near term trends include the continuation of ease of use improvements, away from batch to interactive windowing in native fonts. For example, Tibetan script dictionary databases are beginning to appear. A group of people collaborating on Tibetan OCR has made progress, so a working system is possible in the (hopefully) not too far future. This same group has worked on lexical analysis of the ACIP files to determine expert system parsing rules to drive a neural net recognizer.

Longer-term directions include voice recognition of Tibetan words or spelling. Spelling recognition is technically much simpler, as there is no need for context recognition, and could have application for text input projects. Text to speech of Tibetan should be quite straightforward, and could have use in checking input texts by spelling back what was input while the human reviews the original text. Unicode will allow Tibetan text to be rendered in different fonts without the need for font conversion programs. XML or SGML could be used for markup of Tibetan texts to indicate authors, sections, and references to other works, for example. Tibetan language learning CD-ROMs could incorporate Tibetan text with audio. Finally, the customization of operating systems may evolve and open up to allow individuals to customize it for Tibetan, much like there are now Chinese customized versions of Windows.

In summary, the future of Tibetan computing looks hopeful, although probably limited in the variety of competing products compared to the overall computer software industry.

Appendix

1) Programming Files

The keyboard directory has several files, which include:

WYLIEKB.DLL is the Windows Dynamic Link Library of application callable routines. (See below for more detail.)

WYLIEKB.EXE is the Windows Tibetan Keyboard Driver. This program intercepts user keystrokes via system hooks and passes the characters to functions in WYLIEKB.DLL for translation.

Remaining files are the runtime ObjectVision environment that executes the OVD files, as well as various data and documentation files. If a user has purchased the ObjectVision development package, an entire dictionary application can be created using a point and click interface.

2) Programming Interfaces

If you want to create new Tibetan computer applications, many of the programming routines used in the Tibetan Keyboard and Tibetan Dictionary are available for your use. The following section describes these functions and how to call them.

The file WYLIEKB.DLL is a dynamic link library, which has the application callable routines described below. C language notation is used for argument and return type information (char=character, int=integer, far=large address space, *=pointer).

windows_wylie converts Wylie to Tibetan on a character by character basis, and is called from the keyboard driver application on every key. It takes a single char argument.

WylieKeyboard activates or deactivates the Tibetan keyboard. The first argument is an int where 1 means activate and 0 means deactivate. A second int argument returns the success code.

SetTraceMsg activates or deactivates the keyboard character trace in the wyliekb.exe window. The first argument is an int where 1 means activate and 0 means deactivate. A second int argument returns the success code.

Rob2Wylie converts Robillard Tibetan font strings to either Wylie text strings or Tibetan index strings (which serve as Tibetan alphabetizing keys). The first argument is an int, with 0 indicating generate an index and 1 indicating Wylie. The two following arguments are far char * arguments, the first points to the Robillard string to convert, and the second points to a buffer to write the converted string.

SaveOptions writes the current keyboard settings to a file so that the next time the Wylie keyboard is started the options are also restored. Currently only the keyboard input mode is saved (i.e. either Wylie, phonetic or QWERTY). The function takes no arguments and returns a Boolean.

InstallWriteMode sets the keyboard driver to accommodate different Windows applications. The argument is 0 for ObjectVision (or most other Windows applications), 1 for the MS Write editor, 2 for automatically switching between the previous modes depending on the name of the disk file (WRITE.EXE, WINWORD.EXE or neither) of the application and 3 for MS Word for Windows editor.

3) Wylie Transliteration Keyboard Map

This is a sample of one of the keyboard maps included with Tibetan For Windows. Similar maps exist for the phonetic and Tibetan typewriter modes. In addition, users can create custom keyboard mappings by modifying files in a standard text editor (no programming required).

ALT-F8 to enable the Tibetan Keyboard
ALT-F9 to disable the Tibetan Keyboard