Keyboard input

Ken Zook

October 29, 2012

Contents

1 Introduction

Since FieldWorks stores data in Unicode, users need some way to enter Unicode data from the keyboard. With 100,000 or more code points, it provides a challenge for data entry with a keyboard that has about 47 keys. There are various ways to do this. FieldWorks supports the Microsoft Text Services Framework, MSKLC, and Keyman as typical ways to input data. The InKey program currently available in Beta form (www.InKeySoftware.com) will also work as long as the system language you use is defined in your version of Windows.

Each FieldWorks writing system is set to use a certain Windows keyboard and optional Keyman table. The appropriate keyboard is selected automatically, based on the writing system of the text at the cursor. Users can also change the writing system (e.g., embedding another writing system) by selecting a different one from the writing system combo on the toolbar. The keyboard will change accordingly.

2 Text Services Framework

Recent versions of Microsoft Office and some other applications, including FieldWorks, implement the Microsoft Text Services Framework (TSF). This is a complex system to implement, but it provides advanced support for Input Method Editors (IME) and Keyman. For example, with a Chinese IME, the characters are entered just below the

desired location in text instead of in some remote part of the screen. With Keyman, the context where the cursor is placed is used to interpret additional keystrokes.

3 Standard Windows input languages

Windows provides many standard input languages and Input Method Editors (IMEs) that work with FieldWorks.

1. Go to Start…Settings…Control Panel…Regional and Language Options and click Languages.
2. For right-to-left languages or East Asian languages, check the appropriate box(es). **Tip:** This may require the Windows installation CD-ROM.
3. Click Details, then click Add.
4. Select the desired input language and the keyboard layout/IME, then click OK.
5. In both Text Services and Input Languages *and* in Regional and Language Options, click OK.
6. In the FieldWorks desired Writing System Properties…Attributes, select this input language in the System Language for the keyboard input combo but leave the Keyman keyboard combo empty.
7. Click OK.
   Result: Closes the Writing System Properties. Whenever the cursor is placed in a string in this writing system, FieldWorks automatically switches to that keyboard.

4 System language bar

The System Language bar is helpful when diagnosing keyboarding problems.

1. Go to Start…Settings…Control Panel…Regional and Language Options.
2. Click Languages, select Details, and then click Language Bar.
3. Check all the boxes.
   Result: Shows the language icon in the system tray.
4. Right-click the icon and choose "Restore the language bar".
   Result: Makes the icon visible on your desktop and gives detailed information about what is happening.
5. To put the icon back in the system tray, click Minimize.

Some IMEs have options users can select from the system language bar during operation after they are installed.

Example

If users install the Chinese (PRC) input language with the Chinese (Simplified)–Microsoft Pinyin IME 3.0 keyboard and have the System language bar showing, they will see a button that toggles between Chinese and English when clicked.

- When set to Chinese, as users type Pinyin characters, they appear in a small Pinyin box below the cursor location (e.g., *binguan*) and these are converted to Chinese characters.
- When set to English, it inserts English characters instead of activating the Pinyin box.

It is important to set the various IME options to get the desired behavior. These settings remain in effect as long as FieldWorks is open. Users may need to reset them the next time they open FieldWorks.

5 On-Screen Keyboard

Windows XP provides a useful program when dealing with unfamiliar system keyboards.

1. Go to Start…Run.
2. Type *osk* in the open combo and click OK.
   Result: Brings up a window that stays on top of other windows. It shows the keys for the active keyboard, showing various shift states.

. **Tip:** Use the mouse in the dialog to switch shift states and insert clicked characters at the cursor. You may need to move the mouse over the window to get it to update to show a new keyboard.

6 Microsoft Keyboard Layout Creator (MSKLC)

MSKLC is a program that can be downloaded for free from http://www.microsoft.com/globaldev/tools/msklc.mspx. This program allows you to create a custom keyboard from an existing keyboard, a source file, or from scratch. The resulting keyboard installer can be copied to any computer and installed freely. Once installed, it can be used by most programs through the normal Windows language bar.

MSKLC is more limited than Keyman in that it only accepts a single keystroke, or a dead key followed by a single keystroke as input. A single keystroke can be transformed into 1-4 Unicode code points, while a dead key followed by another key can be transformed into a single code point. For example, you cannot use a vowel followed by diacritics to produce composite accented vowels. However, you can have a diacritic transformed to a combining diacritic which will work as long as your software can handle NFD input (this works with FieldWorks). You can also produce accented vowels by typing a diacritic (dead key) followed by a vowel, assuming there is a composite code point that has the diacritic and vowel.

Fieldworks provides a MSKLC keyboard developed by the NRSI for typing IPA characters. The file to install is c:\Program Files\SIL\FieldWorks\Keyboards\IPA\MSKLC\ipa101\setup.exe.

The following chart makes some comparisons between MSKLC and Keyman.

Features	MSKLC	Keyman
Can map a single key (with various shift states) to any single Unicode value	yes	yes
Can map a single key to multiple Unicode values	yes	yes
Any key can be designated a dead key and any single key following the dead key can be mapped to a single Unicode code point	yes	yes
Can map two or more keystrokes to a single Unicode code point (without the dead key approach)	no	yes
Can map keystrokes to Unicode code points above Plane 0 (the application must also be able to handle Plane 0 and higher)	yes	yes

Context sensitive input	no	yes
Can replace previously output characters	no	yes
Can do reordering, for example so that a prevowel is typed before a consonant but stored after it	no	yes
Can enforce normalization in keyboard	no	yes
Works with: ⤷ SIL FieldWorks, MS Office, OpenOffice, ZEdit, ⤷ Toolbox 1.5, etc.	yes	yes
Works with: Paratext and older ⤷ Toolbox	no	yes
Freely downloadable	yes	no
Use any existing Windows system keyboard as a starting point	yes	no
Functions on Win9x	no	yes

While MSKLC is not as powerful as Keyman, it is adequate for many languages needing custom input and keyboards can be distributed and used freely.

When assigning letters, if you double-click a letter and click All and the Advanced checkbox, you'll be able to assign keys to ctrl+alt+<Key> in shifted and unshifted states. When using the keyboard, holding down the right Alt key will access these keys. You can also assign keys to SGCAP+<Key> in shifted and unshifted states. When using the keyboard, you access these keys with the Caps Lock key in the locked position. If you have both ctrl+alt+<Key> and SGCAP+<Key> assignments made, you can access all of your keys by setting Caps Lock mode. Then you can type a key, shifted key, right+alt key, and shifted+right+alt key to access the four states.

Microsoft Word has a built-in function that allows a user to access standard diacritics by holding down Ctrl+diacritic followed by a vowel. Thus Ctrl+'a will produce á. You can program this functionality in MSKLC by clicking the apostrophe key, then clicking the All button which brings up an assignment dialog. In this dialog, click both the Dead Key View and the Advanced View checkboxes. You can then assign the apostrophe to the Ctrl+<Key> slot and make it a dead key. Then clicking the … button to the right allows you to enter the various vowels and corresponding composite code points.

6.1 Creating an MSKLC keyboard

1. Run the MSKLC program through Start…Programs…Microsoft Keyboard Layout Creator 1.4.
2. It's usually best to start with an existing keyboard that is close to what you want. To do this, use File…Load Existing Keyboard, then choose a keyboard from the list.
3. By clicking the check boxes on the left, you can see the keys assigned to each key state.
4. To change an assignment, click a key on the displayed keyboard and type in the desired letter(s) or Unicode code point(s) (e.g., U+110C). If you want to set or see all shift states, click the All button and it will bring up a dialog allowing you to change the results from each shift state.
5. To set up a dead key, click a key on the displayed keyboard and click All, then in the dialog click the Dead Key View checkbox. Then you can click the Dead Key checkbox for the shift state you want for your dead key. Then clicking the … button will bring up a dialog. From here you enter a base code point (the key following the

dead key), and the composite code point or Unicode value to which it should be transformed. You should normally include the dead key in the final line as a base and composite character. This way by typing the key twice you can insert the key used as a dead key.

6. To test a keyboard, choose Project…Test Keyboard Layout. The keyboard will be in effect as you type in the window.
7. In Project…Properties, you must assign a Language. When the keyboard is loaded, this keyboard will automatically be added to this Windows language. This can be confusing if you leave it defaulting to English (United States) EN. You may find that when you type English the keyboard gets into this mode, giving unexpected results. If you know the system input language you will use with keyboard, set it in this dialog so that it will be assigned automatically. Otherwise, if it shows up under the English input language, you'll want to delete it.
8. To make an install file for the keyboard, choose Project…Build DLL and Setup Package. This will save a directory in My Documents that contains a setup.exe file plus some other files and subdirectories.
9. To install the package, copy the entire subdirectory to the target machine, then run the setup.exe program. To uninstall the keyboard, use Add or Remove Programs.
   Note: After installing a keyboard the keyboard will be added to the system input language specified when the keyboard was designed. This is frequently English (United States). In most cases, you'll want to delete this keyboard from that input language and assign it to your desired language (see next step).
10. To use the keyboard, you must assign it to a system input language and then choose that input language in your editor to activate the keyboard.

6.2 Setting up MSKLC in FieldWorks

Note: In this example, Tester is the name of the MSKLC keyboard and Punjabi is the system input language.

1. Install the Tester MSKLC keyboard by double-clicking the tester\setup.exe program. This keyboard now shows up in the normal list of system keyboards.
2. To make it available to programs, go to Start…Settings…Control Panel…Regional and Language Options.
3. Click Languages, select Details, and then click Add.
4. Select an input language that *will not* be used for anything else (e.g., Punjabi) in the Input language combo.
5. In the Keyboard layout/IME combo, select the installed Tester MSKLC keyboard, then click OK.
6. In both Text Services and Input Languages *and* Regional and Language Options, click OK.
7. In FieldWorks, go to the desired Writing System Properties…Attributes tab, and in the System Language for keyboard input combo select Punjabi. Leave the Keyman combo set to None.
8. Click OK to close the Writing System Properties.

Result: Sets up FieldWorks to use the Tester MSKLC keyboard any time you click in a field using that writing system. If the System Language Bar shows at this point, it will indicate Punjabi.

7 Keyman

Keyman is a program developed and supported by Marc Durdin (www.tavultesoft.com). Earlier versions of FieldWorks provided an option to install Keyman 6.2 that could be registered and used freely by SIL members. Tavultesoft no longer issues new registrations for version 6.2, but those that have previously registered can continue to use it. Tavultesoft now distributes Keyman 7 which has several enhancements including the ability to work with Microsoft Vista. However, it is no longer free. Everyone must pay to use it beyond a 30 day trial period. Keyman 7 can be downloaded from http://www.tavultesoft.com/70/download.php. You can also contact Tavultesoft (sales@tavultesoft.com) for more information.

Keyman is a powerful program that provides many-to-many mapping between keyboard keys and code points. For example, tir<space> can be converted to a single Unicode code point or any number of Unicode code points. It also provides for context sensitive changes. For example, a keyboard may convert an 'a' followed by '`' as an 'a' with a grave accent (à). With TSF support, if you place your cursor after an 'a' and type '`' it will convert these two characters to an 'à'. If MSKLC does not provide sufficient support for your language, Keyman will almost certainly work.

If you are designing Keyman keyboards for use with FieldWorks, you need to be aware that context sections must be designed to expect Unicode characters in Normalization Form Decomposed (NFD). The output of a Keyman table for FieldWorks programs can be NFD or NFC (Normalization Form Composed).

Note: Keyman 7.0 Light, and the former free version of Keyman 6.2 do not support TSF and are limited to two keyboards.

Fieldworks provides a Keyman keyboard developed by the Non-Roman Script Initiative (NRSI) for typing IPA characters. The file to install is c:\Program Files\SIL\FieldWorks\Keyboards\IPA\Keyman\IPAUni11.kmp.

7.1 Known issue

Keyman uses a keyboard hook in Windows to intercept processing of keyboard keys and convert it into other keystrokes based on the selected Keyman table. Sometimes there are problems with this because other programs may also use keyboard hooks. If another program gets to the keystroke first, it can prevent Keyman from working.

Workaround: Make it possible for Keyman to get the hook first or eliminate the other program. In these cases, it often works to have Keyman start when Windows starts. Set the Keyman Options…Configuration by checking "Start Keyman with Windows".

7.2 Creating a keyboard

To develop Keyman keyboards, users must have Keyman Developer (Tike). Edit the source .kmn file for Keyman tables in any ASCII editor or use the Keyman Developer program. Then use Keyman Developer to convert this into a compiled .kmx file.

tester.kmn is a sample Keyman table that handles most European accents. Here is a typical line from a Keyman table that converts an "a" followed by hyphen to an "a" with a macron:

6/13/2013

'a' + '-' > U+0101 c a with Macron

In the past, Keyman keyboards had to handle the context sensitivity for complex scripts which often required complex Keyman tables. With Unicode, the Keyman keyboards can usually be quite simple since the context sensitivity now takes place when rendering the data.

Note: Keyman tables must be designed *specifically* for Unicode, so users cannot use older Keyman tables with Unicode applications such as FieldWorks. In FieldWorks, Keyman works directly in the FieldWorks internal text buffer which is always maintained as NFD normalized data. As a result, Keyman tables for FieldWorks must be prepared to handle NFD data in the context section. The output can be NFD or NFC since FieldWorks will automatically convert it to NFD. For example, the following command will work in most programs:

any(consonantU) U+0CCD + 'M' > index(consonantU,1) U+0CCD U+0C82

However, this will not work in FieldWorks because U+0CCD decomposes to U+0CBF U+0CD5 in NFD. Thus FieldWorks needs the command:

any(consonantU) U+0CBF U+0CD5 + 'M' > index(consonantU,1) U+0CBF U+0CD5 U+0C82

The context to the left side must be NFD, while the replacement on the right can be either NFD or NFC. You could have both of these commands in the same Keyman keyboard to allow it to work in most programs (e.g., Word, Toolbox) that normally use NFC as well as FieldWorks that uses NFD.

7.3 Installing and using a Keyman keyboard

Keyman tables must be installed in Keyman before trying to use them.

1. Double-click the .kmx/.kmp file *or* right-click the Keyman icon and choose Keyman Configuration.
2. Click Install keyboard.

In non-FieldWorks applications, to type using this keyboard, click the Keyman icon and select the desired keyboard. Once FieldWorks is set up properly, the switch will take place automatically without using the Keyman icon.

7.4 Using non-English Keyman keyboards

You can run into complications with Keyman when using non-English keyboards. Keyman defaults to positional layout, using the key locations on a US keyboard. If your table converts "a" + "`" to create an "a" with a grave diacritic and you use this Keyman keyboard on a German keyboard, you have to chose "a" + "^" on the German keyboard. The key to the left of the numeral 1 is ^ on a German keyboard, while it is ` on a US keyboard. When designing a Keyman table, insert

store(&mnemoniclayout) "1"

to use the mnemonic layout. In this mode, press "a" + "`" as shown on the keytop for that keyboard, even though it is in a different location than on a US keyboard.

7.5 Keyman in FieldWorks

FieldWorks automatically selects the correct Keyman keyboard and uses it when users click any string, as long as things are set up correctly. For this to work properly, users *must* assign a unique Windows input language to each Keyman keyboard. This language should not be used for any other purpose. It's OK to use this same system language for multiple FieldWorks writing systems, as long as each one uses the same Keyman keyboard.

If you do not have keyboards set up properly, FieldWorks fails to switch to the correct keyboard when you click in a string using that writing system.

Programs interact with Keyman in different ways, especially when TSF is also involved. It is possible that a given Keyman keyboard will work one way in FieldWorks, another way in Microsoft Word, and yet another way in Toolbox.

The FieldWorks installation includes a Keyman keyboard for typing IPA characters using Doulos SIL or Charis SIL fonts. This is in the c:\Program Files\FieldWorks\Keyman directory. You need to install IPAUni11.kmp in Keyman in order to use it. IPA Unicode 1.0.5 Keyman Keyboard.pdf in the same directory describes its use.

7.5.1 Setting up Keyman in FieldWorks

Note: In this example, Tester is the name of the Keyman keyboard and Punjabi is the system input language.

3. Install the Tester Keyman keyboard in Keyman.
4. Go to Start…Settings…Control Panel…Regional and Language Options.
5. Click Languages, select Details, and then click Add.
6. Select an input language that *will not* be used for anything else (e.g., Punjabi) in the Input language combo.
7. In the Keyboard layout/IME combo, select the installed Tester Keyman keyboard, then click OK.
    Note: If you are using the limited Keyman versions without TSF support, your Keyman keyboard will not show up in this list. In this case, choose US. .
    Note: Tavultesoft Keyman should also show in the Keyboard section of the input language. If not, context sensitivity will not work, and Indic keyboards will likely not work correctly in some Flex contexts.
    Vista Note: Keyman Desktop Professional 7 on Vista requires an extra step to get Tavultesoft Keyman Desktop TSF Addin in the Keyboard section. First, you need to have at least Keyman version 7.0.241.0 or higher for this to work. After you have defined all of the input languages you need, then go to the Keyman configuration dialog. If you are in the FieldWorks Writing System Properties dialog, you can do this by clicking the blue Keyman Configuration link. In the Keyman Configuration dialog, click Addins, then under kmtip click the Configure button, then click OK to the dialog that adds TSF to installed languages.
8. In both Text Services and Input Languages *and* Regional and Language Options, click OK.

9. In FieldWorks, go to the desired Writing System Properties…Attributes tab, and in the System Language for keyboard input combo select Punjabi, then in the Keyman keyboard combo select Tester.
10. Click OK to close the Writing System Properties.

Result: Sets up FieldWorks to use the Tester Keyman keyboard any time you click in a field using that writing system. If the System Language Bar shows at this point, it will indicate Punjabi…Tavultesoft Keyman…Tester, and if you type an "a" followed by a hyphen, it will show up as an "a" with a macron.

As long as you use the Standard Edition of Keyman, if you place the cursor following an existing "a" in your text and type a hyphen, it will also convert the preceding "a" into an "a" with a macron. This is the context sensitivity provided by TSF and Keyman. Without TSF, or using the Home Use Edition, you have to delete the existing "a" first, then type "a" followed by hyphen for this to work.

7.5.2 Sluggish Keyman input with TSF

When using Keyman 6.2, FieldWorks has a problem when using some Keyman tables with TSF. The combination of Keyman 6.2 and TSF causes keyboard input to be sluggish. If your typing gets ahead of the display, FieldWorks attempts to display all of the characters since the last display update. However, when this happens, it can cause Keyman to miss changing some keystrokes, or even reorder the characters, putting changed characters ahead of unchanged characters. This can be especially troublesome if you type fast or if your computer is slow. This problem occurred with the old IPA keyboard (ipaUni10.kmx). If you encounter this problem and are using this older keyboard, switch to the newer IPAUni11.kmp keyboard included with FieldWorks 5.0.

If you encounter this problem in other Keyman tables, the problem can be fixed by modifying the table, inserting the following code.

```
c This is a fix for the problem of fast typing reordering keystrokes
when using Keyman with TSF
store(chars)
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%
^&*()_+-=[]\{}|;:",./<>?' "'`~"

group rules

c This rule to fix the problem of fast typing reordering keystrokes
when using Keyman with TSF must be the last rule in the group.
 + any(chars) > index(chars, 1)
```

The store(chars)… statement should be between the various header statements and the first begin statement (at the start of the program). Only the + any(chars) >… should be at the end of the last group, following all existing rules (as a 'fallback' rule). If your program uses multiple groups, you may need to add it at the end of each group, unless all unmatched keystrokes always flow through to a single group.

7.5.3 Delayed activation of FieldWorks with cs-pinyin

Tavultesoft provides a cs-pinyin.kmp Keyman keyboard that supports 100,000 Chinese Han characters. However, there is a bug in Keyman 6.2. When this keyboard is installed and activated, FieldWorks applications are delayed approximately 30 seconds every time

they are started. After this startup delay the applications run at normal speed. There is a workaround for this problem. Contact Language Software Development in Dallas if you encounter this problem.

8 InKey

InKey is under development in India and a beta version is available at http://inkeysoftware.com/. It is free for SIL use. It has complex capabilities similar to Keyman. When used with FieldWorks, it works similar to MSKLC. The InKey keyboard gets set to a system input language, and then in FieldWorks, you set the System language to the language associated with your InKey keyboard.