# The MicroSoft Keyboard Layout Creator

For keyboarding special characters in Toolbox, my recommendation is to use the MicroSoft Keyboard Layout Creator, MSKLC. This free program creates keyboards like the built-in Windows keyboards. They work with all Windows programs.

Even better than being free, it's quite easy to create a keyboard or modify an existing one. This document describes the process in full detail.

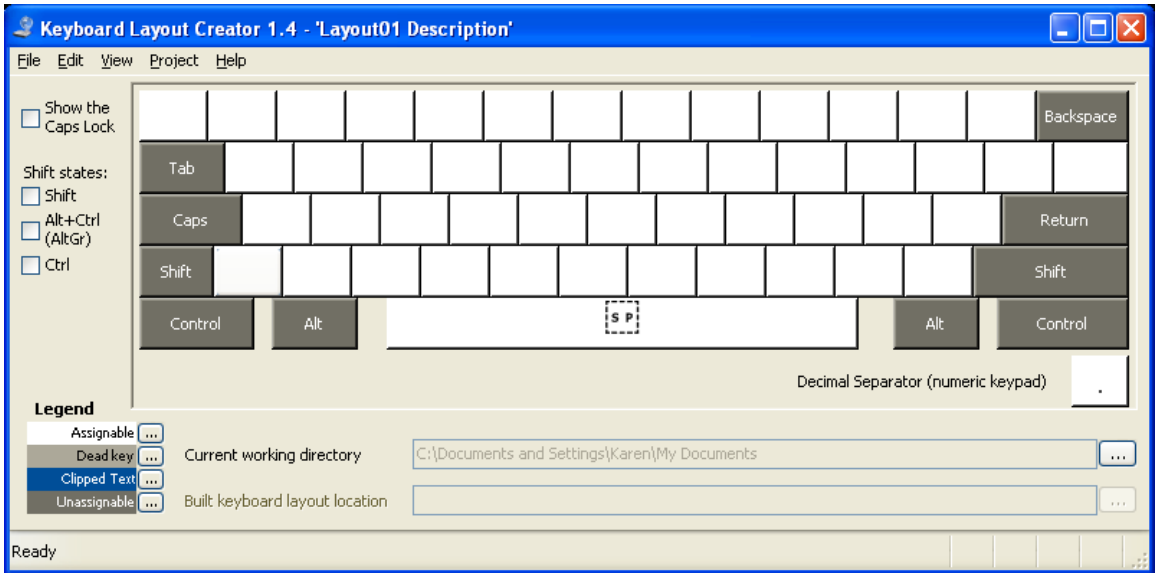
If you Google "MSKLC", you will probably find a link to Microsoft.com at the top of the list. Take that and download version 1.4 (which I believe is the latest).

Download and install the program.

## 1. A Tour of the Layout

### *1.1 The Keyboard*

Run the program. It will open a blank keyboard. It may warn you not all features are available on systems older than Vista. Click OK.



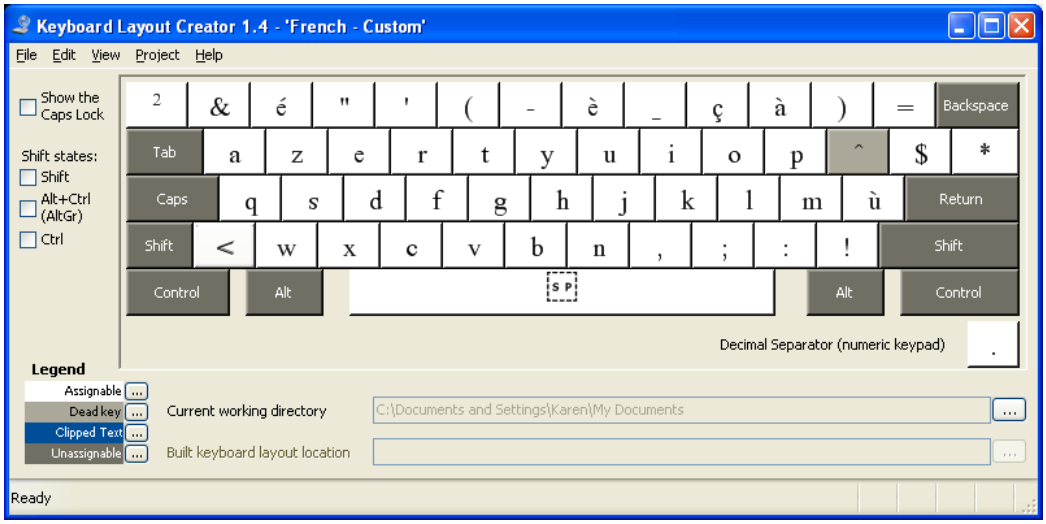

Take note of a couple things I initially missed: Under the keyboard itself, is a location for the "Current working directory" and the "Built keyboard layout location". The "…" at the end of those lines allows you to browse for the location you prefer.

Looking at the keyboard itself, most keys are white. The Legend in the lower left corner indicates this means the key is "assignable" – ie, you can change it. The dark grey Tab, Return, Shift and other such basic keys are "unasignable".

The first thing you should do is load an existing keyboard and look at how things were done. I recommend French, as it has various useful features to use as an example. In the examples below, I have loaded the French keyboard. Do File, Load Existing Keyboard. You'll be given a long list to choose from. You may have to scroll to locate the French.

Looking at the French keyboard (below), you can also see that there are various types of changes that can be made to a keyboard. First, keys can be placed in different locations. The a, z, q, and w (among others) are in quite different locations from the English layout. Also, keys which are not present in the

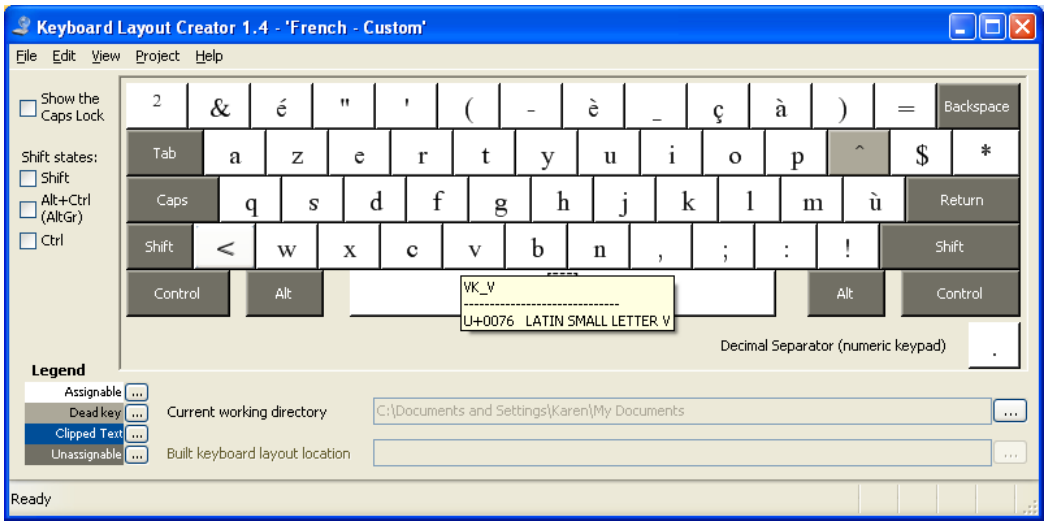
plain English keyboard are available for the French. The top row of keys contains accented vowels and the c-cedilla, for example.



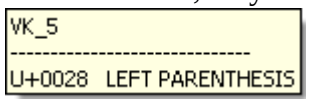

Note, also, the darkened key beside the p (on the right side). Checking the Legend, we see this is a "dead key". This will be discussed in more detail below.

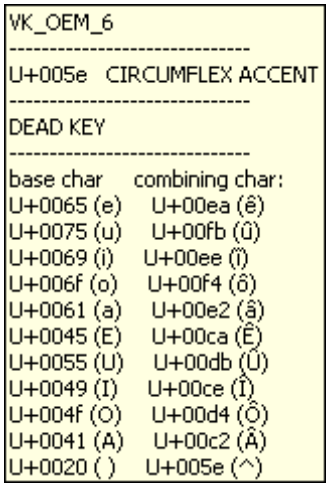
If you click on the Shift checkbox to the left of the keyboard, you will see the capitalized form of these characters. Also, the numbers are on the shifted form of their usual keys. As with English, these are the keys you will see if you hold down the Shift key while pressing the white key. It is similar with the other check boxes. They offer way that combination keystrokes will produce a character. For French, most are empty except the Shift keys. (Undo all the checked boxes.)

Next, notice that if you let your mouse cursor rest on a "key" for a few seconds, a small window will appear with information about the key (in this case, "v".) The "VK_V"




stands for "Virtual Key V" – the official designation of that physical key. As I understand, this is the code the keyboard itself sends when you press that key. The "U+0076 LATIN SMALL LETTER V" is the Unicode name of the character which is actually displayed when that key is pressed. In this case, they are both "v". If you move your cursor straight up, to sit on the left parenthesis, "(", you will see the two specify something different. The box says that the left parenthesis is being shown where the digit five is normally found.






Now let your cursor rest over the dead key (the dark key to the right of the "p"). You will see something rather different.

It starts off with the key designation and then gives the displayed key. That much is the same as we've already seen. Then it says "DEAD KEY". Under that comes an interesting list in two columns. This list says that if you press the circumflex (assumed), and follow it with a base character (first column) of "e", you will see a "combining character" (second column) of e with a circumfles accent on it, "ê". There is a long list of lower and upper case vowels, terminated (customarily) with a space which means if you press circumflex and then a space, you get just a circumflex.

That's what a dead key is. It means that when you press it, nothing happens. It's like it's dead. But then, when you press the next key, you get the result of the two – the accent and the vowel.

As for the key designation, I don't have any idea what the OEM stands for, nor why it's "6". Most, maybe all punctuation characters are "OEM".

Another thing which I don't understand is why the "a" key, for example, which certainly seems displaced to me, claims to be on VK_A. Life is full of mysteries.

However, I've made several keyboards and can assure you that you don't have to understand everything to make this work effectively.

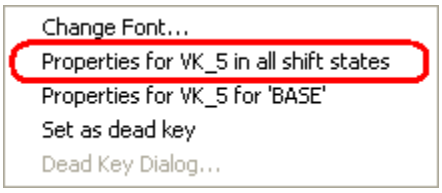
## *1.2 Try It…*

This is usually used after creating a keyboard, for testing, but it's useful for seeing how things work. Do Project, Test Keyboard Layout. You'll be given a big, blank dialog box. Type the "v" key. It should produce a "v". Type the five key, "5" on your keyboard. It should produce a left parenthesis, "(".

Now press the dead key to the right of the "p" – the left square bracket key, "[", on my keyboard. *Nothing should happen*. Now press an "e". You should get a circumflex over the e, "ê". OK, Now press the dead key and press an "a". Oops, did you get a circumflex followed by a "q"? Yeah, I forgot it was moved, too. So press the dead key and the key marked "q"… that should give you a circumfles over an a, "â". When you're done trying it out, click OK to close the test window.

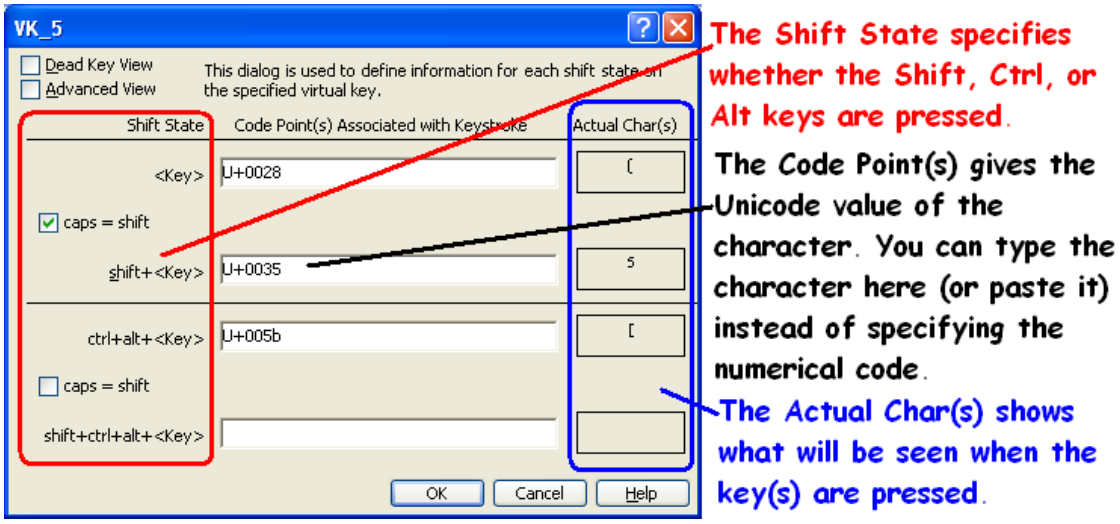
## *1.3 Key Specifications*

### 1.3.1 A Regular Key




One more thing to see is the key specifications and how they are changed. Do a Right Click on the key displaying the left parenthesis, "(". It will pop up a small menu. Choose the second item, "Properties for VK_5 in all shift states". As will become apparent viewing the dialog box, by "all shift states" MSKLC includes the CTRL and CTRL+ALT options as well as the regular Shift key.

The program will display a dialog box like the following:

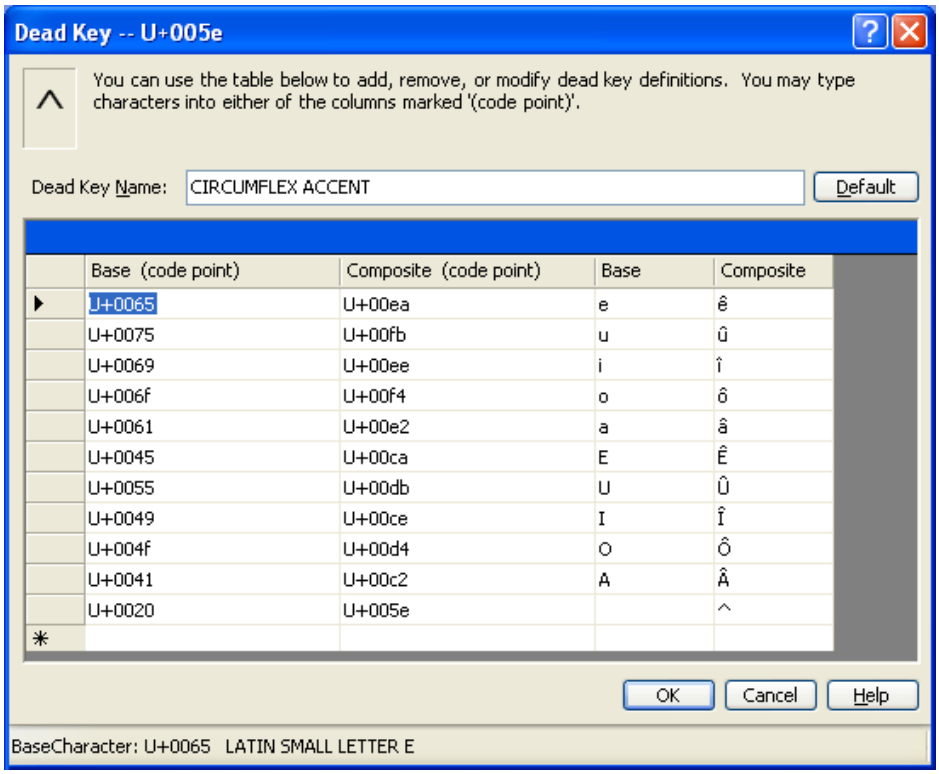



Click on "Cancel" to return to the plain keyboard display.

### 1.3.2 A Dead Key

Now let's take a close look at the dead key to the right of the "p". Again, do a Right Click on the key. Note that the last item on the menu is "Dead Key Dialog". Choose that.

You will see quite a different dialog box appear.

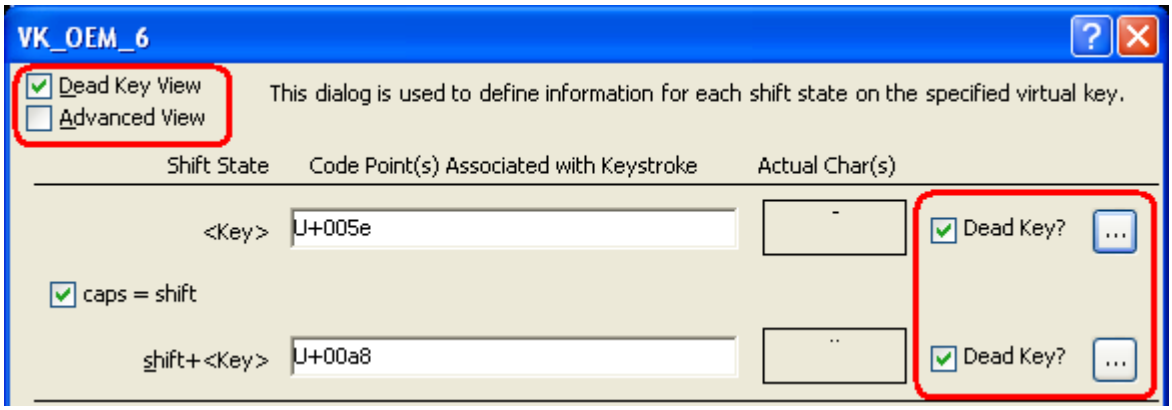



(I shortened the dialog box a bit.)

Note that the key which is displayed on the dead key, the circumflex in this case, is displayed at the top, along with its name. The last line, above the asterisk, specifies a space (U+0020). If you place your cursor on that number, the name will display at the bottom (which now shows LATIN SMALL LETTER E).

There is actually another route to this dialog box. If, when you do the Right Click on the key, you choose "Properties … in all shift states", you will see something like the dialog box that is displayed for the parenthesis. For some reason, the "Advanced View" was checked and that doubles the size of the dialog. Plus it exposes you to the Scan Code, which is something you ***must not change***.

So first, un-check the Advanced View checkbox near the top left of the dialog box. The dialog will then change to look very much like the parenthesis dialog shown above. (The dialog box below has been trimmed to show only the top half.)



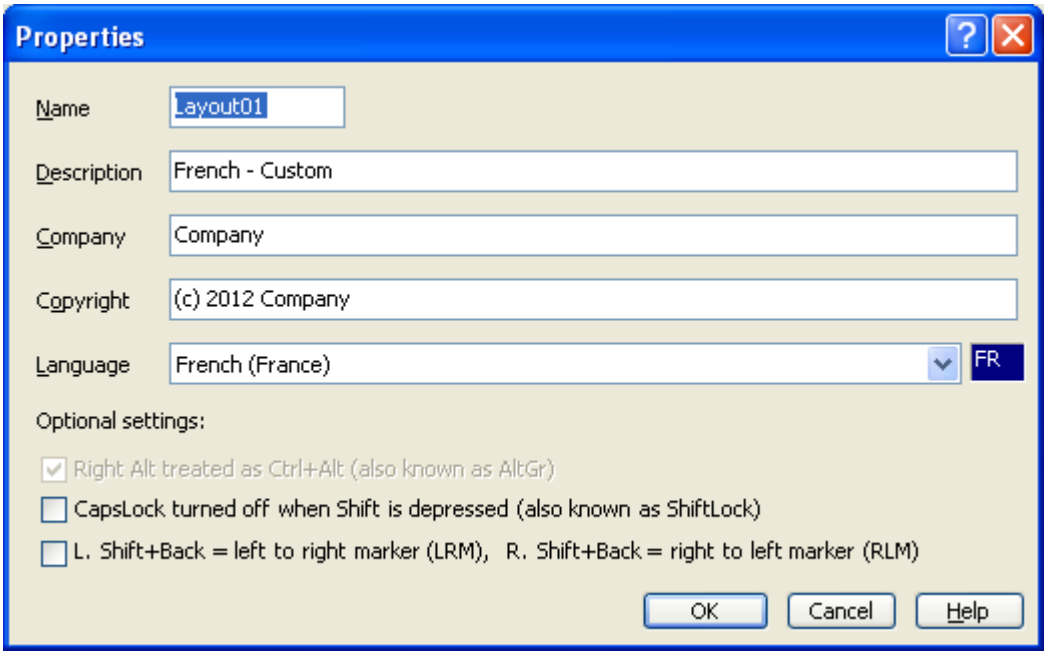

The Dead Key View checkbox in the upper left corner adds the Dead Key checkbox and the three dots, "…", on the right side. If you click the box with the three dots on the un-shifted (first row) key, you will see the Dead Key dialog box which we were just examining above.

## *1.4 Two More Features*

There are just a couple more things you need to see before you dig into making your own keyboard.

### 1.4.1 Project, Properties

The Properties dialog sets some important parameters.




The Name has a length limit of 8 characters and will become the file name.

The Description is the "name" you will see in lists, such as when you loaded an existing keyboard.

The Language is important for a couple reasons. One is that Toolbox selects based on Language, not keyboard name, even though you choose the keyboard name from the list. This means that if you want to have both the plain French keyboard and

a modified French keyboard active (for example, if you have left certain things out of the modified keyboard that you still need for typing French) then you might wich to choose a different Language name (and be sure the abbreviation changes). Note, however, if you do choose a different language that it should be the same type of writing system. You shouldn't choose Arabic for a modification of a European language; nor should you choose French for a right-to-left variant of Arabic. (I'm not 100% sure that this is a still problem, but it makes sense to me. I was told this years ago and it was true then as I discovered "the hard way".)

It would be nice to be able to choose the abbreviation, but that isn't possible. It isn't even possible to see it until you select something, and there are a few surprises.

### 1.4.2 View, Font and Options

The choice of font for the keyboard is made at View, Font.

The View, Options gives you a choice of keyboard layout. As I understand, it doesn't matter but you might as well work on something that looks like what you will be using.

# 2. Making Your Own Keyboard

You've had the grand tour and you are probably thinking that you have a pretty good idea what you might need to do already. And you're probably right. Just a few other comments on the topic.

Your first step should be to load a keyboard very like the one you want to end up with. If you are working where the national language uses the French keyboard, for example, then it makes sense to start with that one. If you are working in India, you might want to start with the keyboard which uses the script you will be modifying. It's much easier to add a few keys than to start from nothing and put everything on the keyboard. (You can do that, however, if you wish!)

Next, go to Properties and choose a file name, specify a Description (which will be the name on the list), and select the Language. If you are just adding a few new things and the keyboard is still useable for the original language also, then you can keep the original language specification (though you will have an extra step below). If you are making differences such that you can't really type French (or whatever) anymore with your modified keyboard, then – if you want both French and the modified French keyboard active – you should choose a different European Language. For example, I modified the US keyboard so that the apostrophe and grave accents became dead keys – they became the overstriking acute and grave accents. But it was still possible to get the apostrophe and the non-overstriking grave. So I left it as "US". On the other hand, the IPA keyboard available from SIL attaches itself to the Icelandic language. This avoids various complications because the keyboard, while still producing the basic alphabet, has a number of modifications. And they wanted it available to folks using more or less any keyboard. I've assumed Icelandic was chosen because the abbreviation starts with "I", like "IPA", but there are other possible reasons.

## *2.1 Choosing Keystrokes*

For the most part, your choices for a character are

- plain keystroke

  Pro: convenient location

  Con: an experienced typist probably expects something else there.

  If you have an extra vowel, for example, to add, you probably want it on the main keyboard. Consider replacing punctuation characters, such as the square/curly bracket keys to the right of

the "p" and the +/= sign and the tilde/grave accent. You might also consider moving the hyphen to a less convenient location unless it is used often. The displaced punctuation can be put on something with less convenient keystrokes. As you saw from the French, it is possible to move the numbers as a group. Remember it is just the keying you are replacing.

- Shift + plain keystroke

  If you've added a vowel, you will want the upper case version here.

- CTRL+ALT+plain keystroke (CTRL+ALT can be Right ALT key)

  The CTRL+ALT combination is awkward, but can be used for the punctuation you've displaced, for example. It is also possible to replace this combination with the Right ALT key (not the Left). However, consider the location of the keys before you do, as pressing a Right ALT with something on the right side of the keyboard would be as awkward as the CTRL+ALT.

- CTRL+plain keystroke

  CTRL is fairly convenient to press, but many programs already make use of the CTRL key. The use of CTRL+C, CTRL+V, and CTRL+X for copy, paste, and cut is almost universal among programs. You don't want to lose them. Likewise CTRL+S for Save. In addition, CTRL+M is Return, CTRL+I is tab. There are others, also. I'm not sure why they made CTRL available as it has a lot of potential for causing problems. Use it with great caution.

### 2.1.1 Dead Keys vs. Live

Dead keys are odd because you press a key and *nothing happens*! It's disconcerting to some people. It is done because the overstrikes in Unicode fonts are designed to overstrike the ***preceding*** character and, apparently some people like to key the diacritic first.[1]

If you don't mind keying the main character first and then the overstrike, you can avoid the use of dead keys. Just put the combining form of the diacritic on whatever key and it will be overstrike whatever precedes it. This saves the complication of having to locate all the combinations.

Note that if you want to use dead keys, make sure the combinations you want exist as you can only specify a ***single*** character code for a dead key. There are acute accents in single-character units over the five latin vowels, but not over open o (ɔ) and open e (ɛ), for example. So different arrangements would have to be made for keying them. It might be better to use a live key.

### 2.1.2 Alternate Forms of Characters

Note that whereas legacy fonts often had multiple versions of a diacritic to look right on main characters which were tall, short, wide, narrow, etc, Unicode does not have that problem. It uses the same acute accent for the i as for the O. Also, you don't need a special "dotless" form of the i so the dot doesn't interfere with the diacritic. These issues are dealt with by special specifications in the font itself. The same is true in other scripts.

Note that roman scripts do have one alternate form which is a potential complication. A character like "acute accent over e" can be a single character or a combination of two characters. This is a hassle and a problem for programs like Toolbox which treat the two as different. ***If you already have data, be sure you know whether the accent characters are single characters or combinations of two.*** (In Toolbox,

[1] I think this is a left-over from typewriters, which couldn't back up and overstrike but which could strike an accent and then not move foreward; the next character could then strike the same place, thus gaining the accent.

you can arrow past such a character. If it takes two arrows to get past (one which doesn't seem to move at all), then you have the combination. If a single arrow goes past, then you have the single character.

Another set of alternates to beware of are the overstrikes. ***Actual overstrikes are called "combining"***, and they are all together in the IPA fonts. However, there is a parallel set, which actually occurs earlier, which are called "modifiers". *Modifiers do not overstrike.*

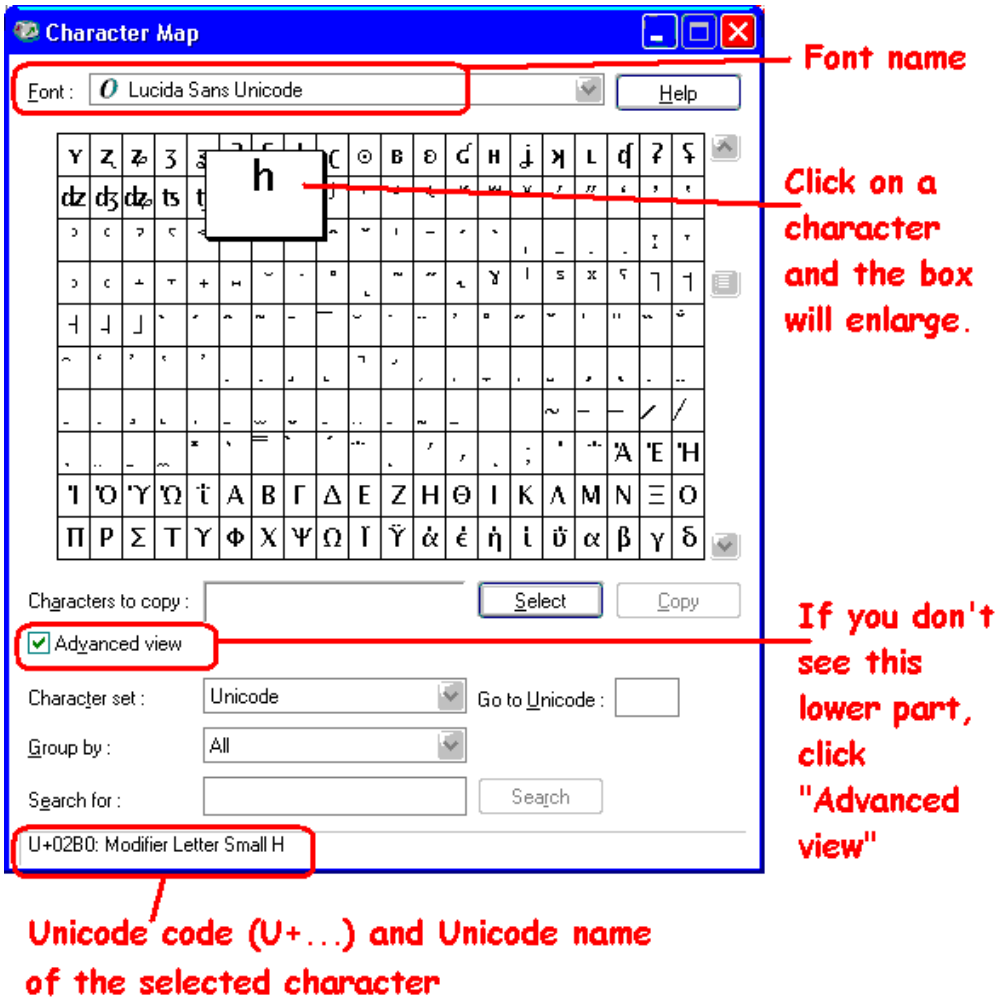
### 2.1.3 Finding the Codes for the Special Characters

I recommend the use of the Character Map for access to the special characters you want to add. The Character Map displays all the fonts which are loaded onto your computer. In addition, you have ready access to any character of that font. The only drawback is that on a small screen, the characters are a bit small.

The Character Map can be found by doing Start, All Programs, Accessories, System Tools, Character Map. It's buried pretty deeply, so you might want to right click on it and choose "Send To" the desktop as a shortcut for easy access.

Select the font you will be working with and be sure the Character Set is set to Unicode. (I didn't highlight that. The picture below was too busy already.) Click on a character to enlarge it. Especially for small items like overstrikes, this is very useful.

Character Map looks like this:



Be sure to read the character name to be sure you have what you want. As I mentioned, "combining" means "overstriking". Modifier characters do not overstrike.

Select a character by clicking on it and then clicking on the Select button. You can then click on Copy. You can select more than one character before you copy.

For the keyboard specification, you can either enter the character itself or the Unicode value (the U+four digits) in the "Code Point(s) Associated with Keystroke". If you enter the character, the MSKLC will convert it to the Unicode value and display the actual character in the "Actual Character(s)" area.

Note also that more than one code can be associated with a single keystroke. It's occurrence in the existing keyboards is rare, but you can put a main character and an overstrike – as two distinct characters – on a single key. You aren't forced to use the combinations. Up to four characters can be specified. So a frequent but complex combination can be made a single keystroke. Technically even a common (short) word could be put on a single key.

## 2.2 Put the Characters on the Keys

You've seen where these go, and we've discussed most of the combinations.

To make an ordinary key into a dead key, click on the "Dead Key View" checkbox in the upper left, and then check the "Dead Key" checkbox beside the character (plus its shift state) which you wish to use as a dead key. Then click on the "…" to open the Dead Key specification dialog box. You assign the overstriking character (eg, the acute accent) to the key in the main location. The dead key dialog is only for the base characters and their specifications. Look again at the French dialog boxes above under "A Dead Key".

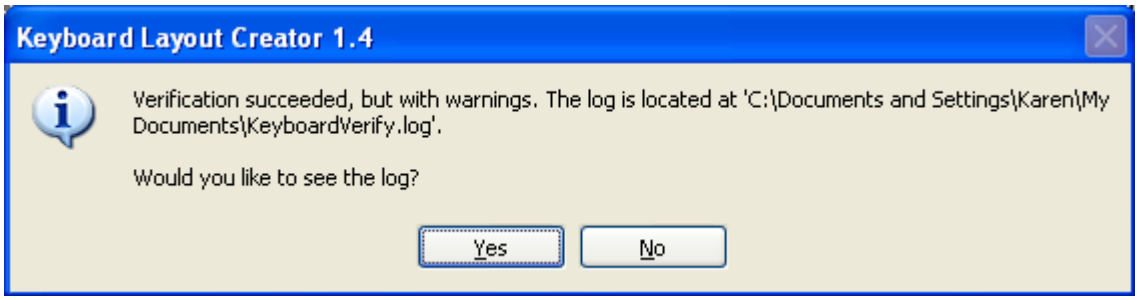
## 2.3 Testing the Keyboard

Again, this is something we've looked at, early in this document. You should probably do a test after you've set up your first key, to be sure it's working the way you think it should. Do it after the first regular key, then again after the first dead key. Be sure to test the whole combination at the end.

## 2.4 Nearly Done – But Verify the Layout

When you believe you are done or nearly done with the keyboard, it is time to Validate the keyboard you've been working on. This is a quick check for various types of errors. This is done by doing Project, Validate Layout.

If MSKLC detects a problem, it will tell you so in the following message box.




It is very convenient to choose Yes here and view the log.

I almost always get some type of message. Usually it is only a warning, but some messages are of a nature that something must be repaired. I don't remember what those situations were at this time. The messages are unusually comprehensible. Here is a sample warning message:

WARNING: v (U+0076) is already defined more than once on the
keyboard (on VK_OEM_3, ShiftState 'Ctrl' and VK_V, ShiftState 'Base').

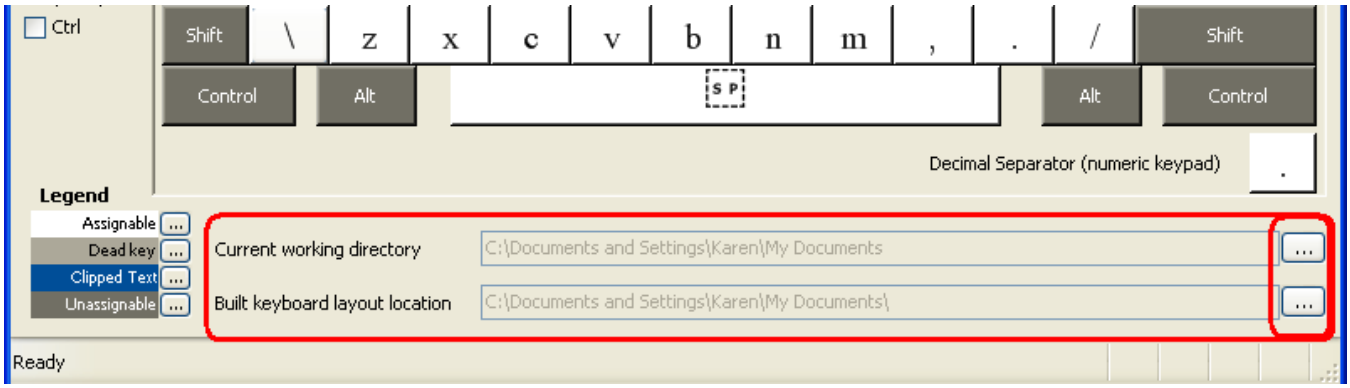
It says I've defined a key twice. I knew that would generate a message because I did that deliberately for one keyboard I made. Warnings can be ignored, though some of them have proved to be quite useful and pointed out things I inadvertently overlooked.

I suggest you do read the log file, even if it says it only contains warnings.

## 2.5 Build the Keyboard Installer

When you have tested and validated the keyboard layout and are satisfied that it does what you want it to do, you are ready to build the keyboard installer.

First, double-check that the working directories and the location for the keyboard are where you want them.



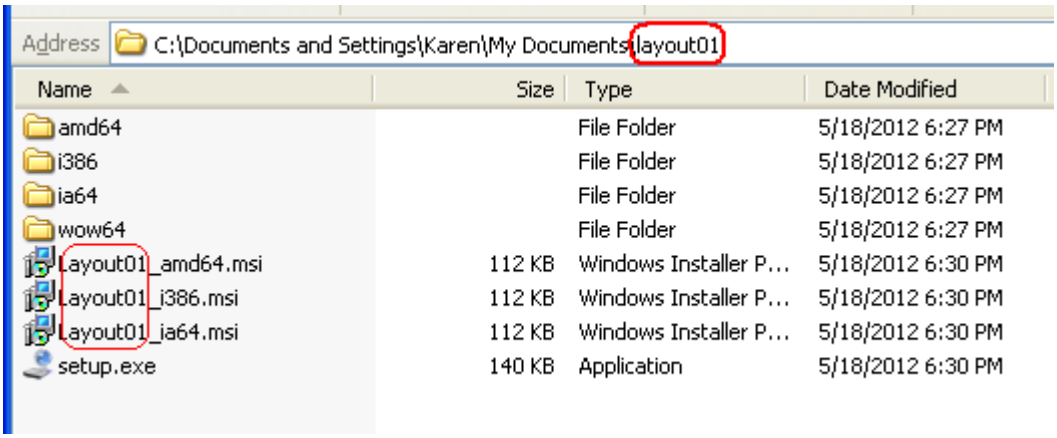

The default location is "My Documents". You can always retrieve them from there if you forget. To specify a different location, click on the three dots, "…" , at the right edge of the keyboard.

Do Project, Build DLL and Setup Package.

> Note that when you click on Build DLL and Setup Package, ***it appears that nothing has happened***. There is no hourglass cursor, nothing. I just built a simple keyboard (the one with the extra "v"), and it took ***over 20 seconds*** before a message came up.

The delay is disconcerting, for me at least.

When it's done, MSKLC will display a message offering to show you the directory. It will look something like this:




The name of the directory is from the first line of the Properties (which clearly I didn't change).

Three possible versions of the keyboard are created, along with four sub-folders. Ignore all that stuff, but don't delete anything.

### *2.6 Installing the Keyboard*

To install the keyboard, double-click on the Setup.exe file

> Again, it will appear that nothing is happening. Be patient. MSKLC gives a message when the keyboard has been successfully installed.

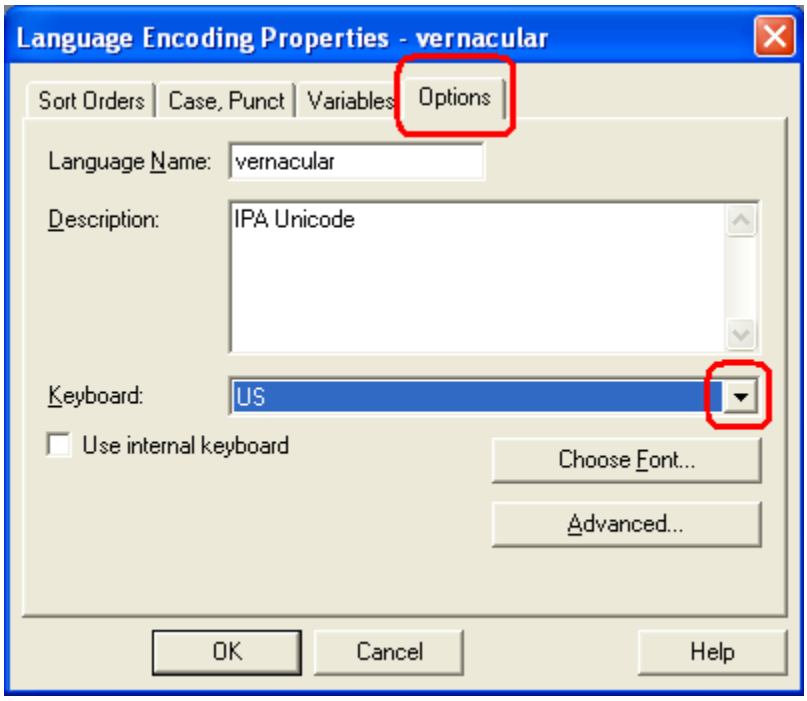
The keyboard will have the name given in the Description of the Properties. It will be associated with the Language that was specified, which determines the two-letter abbreviation the Language Bar will display.

If you wish to send the keyboard to a co-worker, you should give the entire folder ("Layout1" in my sample above.) Your co-worker should also run the Setup to establish the new keyboard.

## 3. Using the Keyboard in Toolbox

To use the keyboard in Toolbox, do Project (in Toolbox), Language Encodings. Toolbox will display a list of all available language encodings for your project. Choose the language encoding you designed the keyboard for.

Click on Modify, and choose the Options tab.




Click on the Keyboard – either the name of the current (probably default) keyboard, or the small arrow at the end in order to drop the list of available keyboards. Choose the keyboard you have created.

Toolbox will apply the keyboard to all fields which use that language encoding – turning on the keyboard when you are in appropriate fields and turning it off when you go to a different field.