

BrianHetrick.com

Application Note AN-1.0

A Latin-1 and Latin-3 Characters US Keyboard Layout for Microsoft® Windows

Goal

The goal of this Application Note is to describe the construction of a Windows 2000/XP/Vista keyboard layout supporting the entry of English and Esperanto language text by persons already familiar with the standard US keyboard. The keyboard layout is built using the Microsoft Keyboard Layout Creator program.

Introduction

Letters, Diacritical Marks, and Language

Most English speakers learn, early in primary school, “the alphabet:” the sequence of letters “A B C D E F G H I J K L M N O P Q R S T U V W X Y Z.” These letters (and their lowercase equivalents) are presented as the symbol set used in written representations of English words. In secondary school, the diphthongs (doubled vowels) Æ and Œ might be introduced. These find use at the secondary school level primarily in the spelling of “encyclo

ædia” and “Œdipus.”

Finally, in late secondary school or tertiary school, English speakers might be introduced to the diaeresis. This double dot, similar in appearance to the German umlaut, is a native English diacritical mark. The diaeresis marks the second of adjacent vowels which are to be pronounced separately. Words and names such as “Cloë,” “coöperate,” “naïve,” “Noël,” and “reëlect” are properly spelled only with the diaeresis. Omitting this diacritical simply misspells these words, just as including it would misspell “house” or “cue.”

Finally, students of English recognize English freely adopts words from other languages, and these words generally keep their original orthography for a few decades or centuries. This orthography frequently includes accents or other diacritical marks. The English

words “façade,” “jalapeño,” “résumé,” and “rôle” are all examples of this. “Rôle” has been fairly well naturalized and, post-19th century is almost always seen without its circumflex; using both accents in “résumé” is sometimes seen as pretentious, but keeping the second is common and will be necessary until the word’s spelling naturalizes; and “façade” and “jalapeño” are almost always encountered with their diacriticals intact.

The unadorned letters “A” through “Z” are not enough to represent most languages — including, as it happens, English.

Character Repertoires, Codes, and Encodings

The computer era quickly saw the computer move from a calculation device to a symbol manipulation device. Many of the symbols manipulated were in letters: names, addresses, words, and so forth. As computers became more common, data interchange became a need: data produced by one computer needed to be consumed by a different computer. This in turn required a common way of representing both numbers and symbols, and the idea of a standard character code was born.

One of the earliest standard character sets was ASCII. First adopted by ANSI as American National Standard X3.4 in 1968, ASCII defined a number of control characters, used for formatting, and graphic characters, used to represent printing symbols such as letters. ASCII defined codes for upper and lower case letters without accents. Later versions of the ASCII standard also defined *encodings* (but not separate codes) for accented characters: for example, “é” could be represented as the three-character sequence apostrophe, backspace, lowercase letter e. However, these multi-code encodings were generally ignored.

© 2008 Brian Hetrick. May be used and copied in accordance with the terms of the Creative Commons Attribution-Noncommercial-ShareAlike 3.0 United States license. The text of this license is available at: <http://creativecommons.org/licenses/by-nc-sa/3.0/us/>.

Variations on ASCII were used for other geographies, which substituted different graphic characters for some of the codes. Eventually these variations were standardized in 1972 with the ISO 646 character code. ISO 646 reserved a number of character positions for “national use.” ISO 646 was probably the first standard to draw a distinction, although implicitly, between the character *repertoire* (the characters that were represented) and the character *codes* (the values that represented the characters).

But information interchange was needed across national boundaries, as well as inside single countries. The ECMA defined a number of 256-code character codes, known as Latin-*n* where *n* is a small integer. These were then adopted as international standards as ISO 8859-*n*, where again *n* is a small integer (but not necessarily the same integer as in the ECMA name).

Although ECMA Latin-1 and ISO 8859-1 (which are identical in their definitions of graphic characters) defined a subset of the Latin letters useful for Western Europe, the 256-element code space was too small to give codes even to all the variations on the Latin alphabet in use. Most recently, the Unicode Consortium has defined a much larger code space, with the goal of giving a single code to each of the characters used in any human language. This code set, Unicode, includes all Latin-1 characters at the same code values as in Latin-1. Unicode is now used by Microsoft® Windows and Microsoft’s layered products on Windows, and is starting to be widely supported on other operating systems.

Latin-1 is sometimes confused with the “Windows character set,” or “code page 1252.” This code, introduced in Windows 3, has graphic characters in the area Latin-1 uses for additional control characters. All these additional graphics are present in Unicode, although at other code points.

Choice of Character Repertoire

The goal of the project described by this Application Note is to construct a keyboard layout for the entry of English and Esperanto text.

As has been seen, English text typically needs more than the unadorned Latin letters ASCII provides. Most current words in at least American English undergoing naturalization come from Western European languages, primarily French and Spanish, due to the developmental history of American English. For these

languages, the characters present in the Latin-1 code are generally sufficient.

On the other hand, while Esperanto vocabulary largely consists of Western European language cognates, its orthography is reminiscent of Southern European languages. Esperanto uses a number of characters that are not represented in the Latin-1 code. These characters are, however, represented in the Latin-3 code. Therefore, having the keyboard layout also support the characters present in the Latin-3 code seems prudent.

One might question whether the keyboard layout should support even more—such as all of the original Latin-1 through Latin-4, or even all Unicode characters based on the Latin alphabet. While this is technically feasible, such a layout would almost certainly imply a large training effort for its users. Consider the variations on the lowercase letter “u” present in Latin-1 through Latin-4 alone:

SMALL LETTER u
 SMALL LETTER u WITH ACUTE ACCENT
 SMALL LETTER u WITH BREVE
 SMALL LETTER u WITH CIRCUMFLEX ACCENT
 SMALL LETTER u WITH DIAERESIS
 SMALL LETTER u WITH DOUBLE ACUTE ACCENT
 SMALL LETTER u WITH GRAVE ACCENT
 SMALL LETTER u WITH MACRON
 SMALL LETTER u WITH OGONEK
 SMALL LETTER u WITH RING ABOVE
 SMALL LETTER u WITH TILDE

Each of these would need to be represented by some “natural” combination of keys. Of these 11 variations, only 6 are present in Latin-1 or Latin-3. This seems a more manageable number.

Therefore, the characters represented in the Latin-1 and Latin-3 codes are sufficient, or almost so. The ligature “œ” and its capital equivalent are also needed to represent English.

Letters, Characters, Keyboards, and Shifts

English and other European languages are somewhat unusual in that they use case in writing. Each letter has two forms, which are used in different circumstances. Most languages are written without case, although some have more than two cases. European languages (and Latin before them) were also written without case, although there were two widely used styles of letter forms: the engraved and the hand-written.

A Latin-1 and Latin-3 Characters US Keyboard Layout for Microsoft® Windows

The engraved, or majuscule, letter forms developed into what are now capital letters. When smooth paper (as opposed to papyrus) became available, rounded uncial forms developed, from which what are now called minuscule letter forms arose. Using both majuscule and minuscule letter forms together started with the development of Carolingian minuscule under the direction of Charlemagne. By roughly the 13th century, mixed case had largely displaced single case and uncial forms throughout Europe. By the end of the 17th century, English usage norms for the letter cases had evolved away from capitalizing nouns and most pronouns, and had settled down to essentially what they are today.

While “A” and “a” are the same letter, they are different symbols, used in different circumstances. This distinction has greatly affected the technology of text. The terms “uppercase” for capital forms and “lowercase” for minuscule forms come from hand typesetting, where capitals and minuscules were kept in separate cases, one over the other. Early typesetting machines used different keys for the different letter forms, associating only one symbol with each key. The first typewriters also had a unique key for each symbol, but arrangements using the now-familiar “Shift” key to cause the production of a different symbol quickly arose.

The role of the *Shift* key is to cause other keys on a keyboard to produce alternate graphics for the same letter, or alternate characters. Keys normally producing lowercase forms produce uppercase forms, while keys normally producing numbers produce symbols, and keys normally producing symbols produce different symbols. A variant on Shift, the “Caps Lock,” causes only letters to produce their alternate graphics. Shift affects only those keys pressed while Shift is held down: the effect of Shift ceases as soon as the key is released. Caps Lock, however, enters or exits a *mode*: its effect lasts until explicitly removed.

By convention, shift keys which cause alphabetic keys to produce symbols, rather than different forms for the same letter, are called “alternate graphic shift” (*AltGr*) keys. This is a misnomer: regular Shift causes keys to generate alternate graphics, while AltGr causes keys to generate alternate characters. Although AltGr is a misnomer, it is widespread and so unlikely to change any time soon.

The use of Shift and an AltGr key together provide four shift alternatives for each key: one with neither

shift used, one for each of the two shifts used individually, and one for both used at once. Various symbol-intensive computer systems have found even this insufficient, and have used additional shift keys labeled “Alt,” “Meta,” and so forth. Each shift key doubles the number of meanings a single key can have.

Shift keys do not need to produce an alternate graphic character. The “Ctrl” key instead produces an alternate non-graphic character. *Ctrl* is used in conjunction with other keys to produce “control characters.” These characters have various special meanings to various computer systems, but generally have no visible representation. Ctrl can be treated as just another shift—combining with Shift and AltGr to produce eight alternatives for each character key—but this is regarded as poor practice.

Although a variety of shift keys is useful for producing a variety of characters, the various shift combinations are not necessarily particularly memorable. A variety of forms of “a” might be necessary: “a,” “à,” “ã,” “â,” “ä,” “å,” “á,” and the corresponding uppercase forms, for example. (Whether these are considered forms of a single letter, or separate letters, depends on the language in use. In French, for example, accents are ignored in “alphabetical ordering,” but in Danish the alphabet ends “x, y, z, æ, ø, å.”) Rather than handle these variations through shift states, it is desirable to indicate the combination of a base letter and an accent. This is done through what are called dead keys.

A *dead key* is a key that modifies the next character entered. These keys represent diacritical marks that are placed on the letter represented by the next key pressed, if that is possible. The character “é,” for example, would be entered as an acute accent followed by the letter “e.”

A symbol generated through AltGr can be a dead “key.” For example, the caron “” (an inverted circumflex) might be generated by Shift-AltGr-6 (that is, AltGr applied to circumflex). Caron could itself be a dead key which when followed by “c” would result in “č” (that is, “Latin Small Letter C With Caron”). This technique is not used in the keyboard layout described in this Application Note, but only because it is inapplicable to the particular characters needed.

The traditional approaches to representing more characters and letter forms than there are keys on a keyboard are therefore shift keys of various types (particularly Shift and AltGr) and dead keys. The keyboard

layout described in this Application Note uses both of these techniques.

The Microsoft Keyboard Layout Creator

Microsoft Windows has always permitted the user to select among keyboard layouts. Even MS-DOS permitted this, at least since version 5 if not before. In all of these systems, developers could design and implement their own keyboard layouts. Keyboard layouts were, however, quite tedious to program correctly. Microsoft has developed and released a keyboard layout creator program, currently at version 1.4. This free download can be used to create keyboard layouts for 101- and 102-key keyboards for at least Windows 2000 onward.

The program provides an easy to use GUI allowing keyboard definitions using dead keys and Ctrl, Shift, AltGr, and Caps Lock shifts. Characters may be defined using either exemplars or Unicode code points. The resulting keyboard layout can be shown in any shift state, and a DLL implementing the keyboard layout can be easily generated.

The Microsoft Keyboard Layout Creator was used to create the keyboard layout described in this Application Note.

Design of the Keyboard Layout

Required Characters

As previously considered, the character set to be supported by the keyboard layout consists of the characters present in Latin-1 and Latin-3, with the addition of the letter “œ” (in capital and small variations).

Latin-1 and Latin-3 share the lower half of their graphic character set with ASCII. These graphic characters are shown in the following table:

	Column					
Row	2	3	4	5	6	7
0	SP	0	@	P	`	p
1	!	1	A	Q	a	q
2	“	2	B	R	b	r
3	#	3	C	S	c	s
4	\$	4	D	T	d	t

	Column					
Row	2	3	4	5	6	7
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(8	H	X	h	x
9)	9	I	Y	i	y
10	*	:	J	Z	j	z
11	+	;	K	[k	{
12	,	<	L	\	l	
13	-	=	M]	m	}
14	.	>	N	^	n	~
15	/	?	O	_	o	DEL

The characters labeled SP and DEL are space and Delete, respectively. The missing columns 0 and 1 hold control characters, which have no standard graphical representation.

The Latin-1 additional graphic characters are shown in the following table:

	Column					
Row	10	11	12	13	14	15
0	NBSP	°	À	Đ	à	đ
1	ı	±	Á	Ñ	á	ñ
2	¢	²	Â	Ò	â	ò
3	£	³	Ã	Ó	ã	ó
4	¤	´	Ä	Ô	ä	ô
5	¥	µ	Å	Õ	å	õ
6	¦	¶	Æ	Ö	æ	ö
7	§	·	Ç	×	ç	÷
8	¨	¸	È	Ø	è	ø
9	©	¹	É	Ù	é	ù
10	ª	º	Ê	Ú	ê	ú
11	«	»	Ë	Û	ë	û
12	¬	¼	Ì	Ü	ì	ü

A Latin-1 and Latin-3 Characters US Keyboard Layout for Microsoft® Windows

Row	Column					
	10	11	12	13	14	15
13	SHY	½	Í	Ý	í	ý
14	®	¾	Î	Ð	î	ð
15	-	¿	Ï	ß	ï	ÿ

The characters labeled NBSP and SHY are non-breaking space and soft hyphen, respectively. Note how the positions of these characters correspond to those of SP and hyphen in the ASCII graphic character set. In the ECMA Latin-1 character set, the missing columns 8 and 9 are undefined. In the ISO 8859-1 character set, these columns hold more control characters, which have no standard graphical representation.

Finally, the Latin-3 additional graphic characters are shown in the following table:

Row	Column					
	10	11	12	13	14	15
0	NBSP	°	À		à	
1	Ĥ	ĥ	Á	Ñ	á	ñ
2	˘	²	Â	Ò	â	ò
3	£	³		Ó		ó
4	¤	´	Ä	Ô	ä	ô
5		µ	Č	Ĝ	č	ĝ
6	Ĥ	ĥ	Č	Ö	č	ö
7	§	·	Ç	×	ç	÷
8	¨	¸	È	Ĝ	è	ĝ
9	İ	ı	É	Ù	é	ù
10	Ş	ş	Ê	Ú	ê	ú
11	Ě	ě	Ë	Û	ë	û
12	Ĵ	ĵ	Ì	Ü	ì	ü
13	SHY	½	Í	Ů	í	ů
14			Î	Ŝ	î	ŝ
15	Ž	ž	Ï	ß	ï	·

The codes corresponding to empty cells in the above table have no defined character in Latin-3. Note how

several characters are shared with Latin-1, and have the same code positions as in Latin-1.

Enabling access to a few additional characters is in order. The ligature “œ” and its capital “Œ” have already been mentioned. Left and right typographical quotation marks (sometimes called “smart quotes”) in both single (‘ and ’) and double (“ and ”) variations are also needed to properly represent English. Although Latin-1 provides double guillemots (« and »), single guillemots (‹ and ›) are absent; both are needed for the same reason as single and double quotes. The lowercase letter “ÿ” is in Latin-1, but its uppercase equivalent “Ÿ” is not; for symmetry we add this letter. The remaining single-case letter, “ß,” does not have an uppercase equivalent. Finally, a character that did not exist when Latin-1 and Latin-3 were being standardized has since acquired substantial importance: the Euro “€.” It seems reasonable to add this character, especially for a keyboard rich in what may be regarded as European characters.

The addition of these 10 characters to the characters of Latin-1 and Latin-3 completes the enumeration of the graphic characters which the keyboard layout is to generate.

Use of Dead Keys and Alternate Graphics

As previously noted, Microsoft Keyboard Layout Creator provides two methods of placing multiple symbols on a single key: the use of dead keys (prefix characters) and the use of alternate graphics (pressing the right Alt key). Many keyboard layouts generating “foreign” characters use both techniques, and in particular the US International keyboard layout provided by Microsoft uses both techniques. However, the US International keyboard layout uses alternate graphics for some alphabetic characters (such as AltGr-Z for “Æ”) that cannot be composed with dead keys, and provides both dead key sequences and alternate graphics sequence for many characters. In general, it seems more satisfactory to use dead keys for letters, and alternate graphics for symbols. This approach is taken with the keyboard layout described in this Application Note.

Choice of Dead Keys

The accents needed in Latin-1 are grave “`,” acute “’,” circumflex “^,” ring above “ˆ,” diaeresis or umlaut “¨,” tilde “~,” and cedilla “¸.” In addition there are the letters “ð,” “æ,” “ø,” “þ,” and “ß.” Latin-3 has most of

these same needs, along with dot above “,” breve “˘,” and line (as in “h”).

The choice of dead keys for most of these is obvious. Circumflex, grave, and tilde are rarely used characters already present on the keyboard; they can represent themselves. In many fonts, the apostrophe looks quite similar to an acute, which suggests its use for that accent. For ring above, any symbol suggesting a circle above the baseline could serve; asterisk seems suitable for this purpose, as it generally rides higher than the obvious alternative, at sign (“*” vs. “@”). For diaeresis, there are two obvious choices: colon and quote. The quote seems a better choice as it was frequently advised as a diaeresis before the word “keyboarding” replaced “typing,” and as quite likely colon is more needed in most text.

Breve is needed only for the letter “u,” and “u” does not need a version with tilde. The tilde has a portion which looks somewhat like a breve, and once again its shape is arguably mnemonic.

The obvious choices for a dead key to represent lines of various types are hyphen, underscore (line below), and slash. Of these, the one least used in most text is probably slash, the choice made. Slash is clearly appropriate for ð and ø, as well. Looking at the ligature Æ, slash is mnemonic for the slanted bar of the A as well. Having lowered our standards this far, it is but a short step to looking at Æ and seeing the asterisk as mnemonic for the round O.

The obvious choice for a key to represent cedilla is the comma. However, the prospect of having to type an extra space after each comma, to avoid it being interpreted as a dead key, is unpleasant. This indicates another choice might be wiser. Slash seems acceptable, so the slash will therefore also serve as a cedilla.

Finally, the letters ß and þ need representations. The sharp s might be considered an accented s; the acute accent seems sensible here. The soft “th” sound of the Old English thorn is fossilized into y in constructs such as “ye olde gifte shoppe” and “hear ye,” so a slashed y becomes a thorn þ.

Dead keys for all the accents needed for the chosen character repertoire have now been identified. These dead keys are grave, acute, quote, circumflex, tilde, asterisk, and slash.

Choice of Alternate Graphics

In addition to a number of letters, the characters the keyboard layout is to generate includes a number of symbols. These symbols are as shown in the following table:

€	¢	£	¤	¥	1	2	3	ª	º
¨	-	'	·	¸	°	˘	©	®	¶
¡	¿	·	µ	¼	½	¾	±	×	÷
¬	¡	“	”	'	'	«	»	<	>
°	§								

These 42 symbols need to become the AltGr versions of keys already on the keyboard; further, it is desirable for these assignments to be mnemonic in some way.

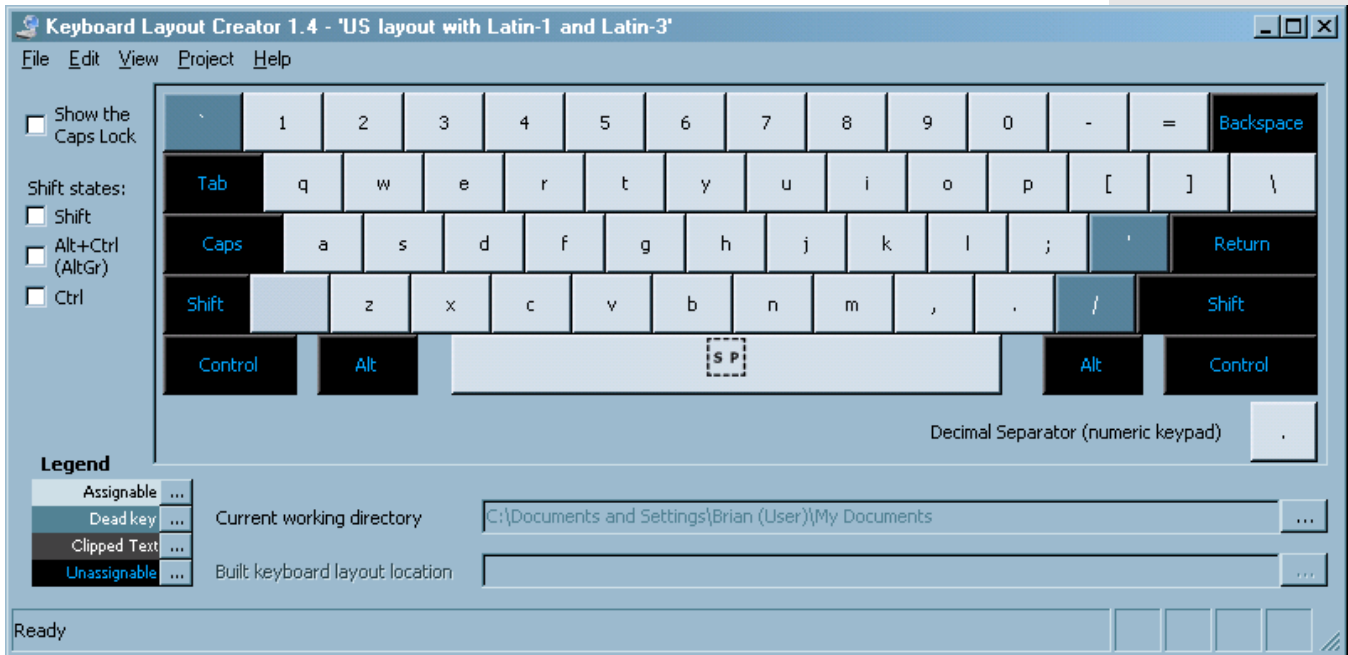
A number of the symbols are either variants of letter forms or based on letter forms. The cent sign ¢ therefore becomes AltGr c; £, L; ¥, Y; §, s; ©, C; ª, a; ®, R; µ, u; ¶, P; º, o; and €, E. This leaves a number of symbols that are variations on symbols already on the keyboard: ¡ becomes AltGr !; ¤ (the international currency symbol) becomes AltGr \$; ¸, |; «, <; ¬, -; ¯, _; °, O; ², 2; ³, 3; ´, ';; ¸, ;, /; ¹, 1; »; >; ¿, ?; ×, x; ÷, ::; ˘, ~; <; (; and >,). Finally, there are a few symbols that are not particularly memorably encoded as variants of symbols: ¼, ½, and ¾ as AltGr 4, 5, and 6; “, ”, ', and ' as {, }, [, and]; and < and > as (and).

Defining The Layout With MSKLC

Microsoft Keyboard Layout Creator supports defining keyboard entry of characters in two ways: using shifts (Shift, AltGr, and possibly Ctrl) and using dead key sequences. These two techniques have different views and activities associated with them.

The simpler of the two methods is defining the results of key presses with the various combinations of shifts applied. This is done by selecting keys on an image of the keyboard and entering characters for those keys. The keyboard image then shows the character generated by the key. An example of this view is shown at the top of the next page. Note how the supported shift states (shift, AltGr, and Ctrl) are shown on the left of the form. Checking the boxes to indicate a set of shift states causes the view to show the characters generated by the various keys when that combination of shifts is in effect.

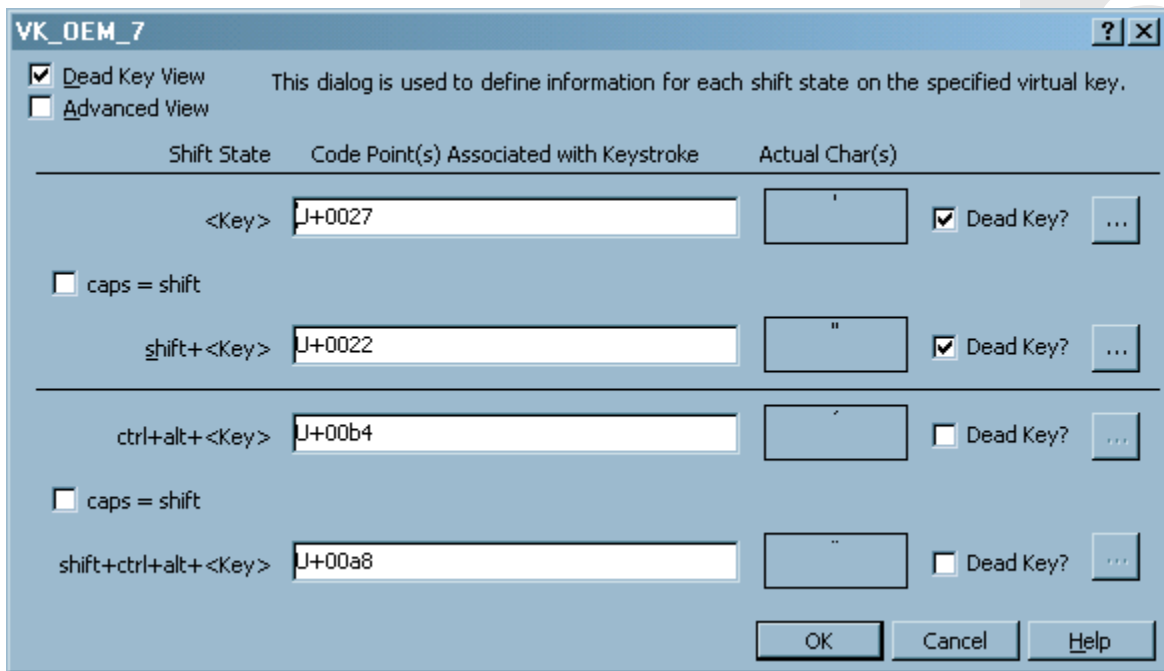
A Latin-1 and Latin-3 Characters US Keyboard Layout for Microsoft® Windows



The keyboard view gives an overview of the key assignments, but is not the form used for making the key assignments. Clicking on a key image in the keyboard view brings up a form like that shown at the bottom of this page. This form shows the character assignments for the key for each possible combination of shift states. (The “Advanced View,” when selected, adds Ctrl to the shift keys used.)

The particular form shown illustrates the assignments for the apostrophe key. When unshifted, the key generates an apostrophe; when shifted, the key generates a quotation mark; the AltGr version of the key is a spacing acute accent; and the shifted AltGr version of the key is a spacing diaeresis.

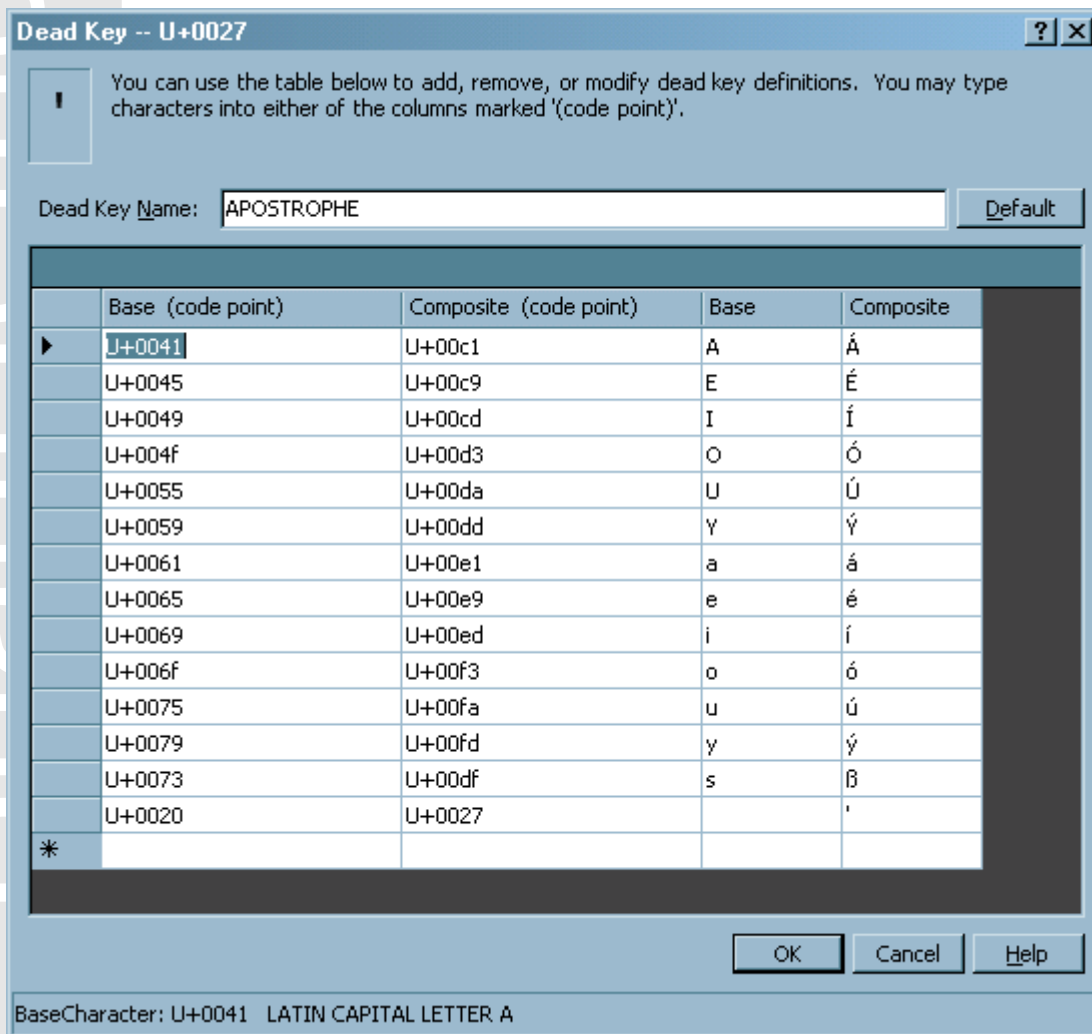
Note how both apostrophe and quotation mark are marked as being dead keys. The additional information button (marked with an ellipsis) brings up the



dead key definition view, shown at the bottom of this page. This view shows the various characters that may be entered following the dead key, and the various characters the keyboard layout is to generate from the combinations. The program allows the entry of either characters or Unicode code points (the letter U, followed by a plus sign, followed by four hexadecimal digits) to specify the entered and resulting characters.

The particular dead key definition shown illustrates the dead key definitions for the apostrophe. Apostrophe followed by an "A", for example, yields "Á," while apostrophe followed by an "s" yields "ß." Dead key definitions should include the combination of the dead key followed by a space being the dead key itself; otherwise, the keyboard layout has the curious property that it cannot generate a single instance of the character assigned to the dead key itself.

The Microsoft Keyboard Layout Creator program can optionally *validate* the keyboard layout, which checks for a number of typically undesired conditions (but not, unfortunately, for duplicate dead key sequences); and it can generate an installation kit for the keyboard definition. It also contains a test facility, where it presents a text box that acts as if it were using the keyboard definition. This lets the keyboard layout be tested to ensure it is as desired, without having to generate and install it. This is quite useful, as actually removing (as opposed to unselecting) a keyboard layout requires a reboot of the machine from which the keyboard layout is to be removed, and attempting to replace an installed keyboard layout can be tricky. To avoid these circumstances, the Microsoft Keyboard Layout Creator program refuses to produce an installation kit for a layout that is already installed on the machine



A Latin-1 and Latin-3 Characters US Keyboard Layout for Microsoft® Windows

Using The Keyboard Layout

Alphabetic Characters

A number of characters normally associated with symbols are used to indicate diacritical marks. These characters and their use are:

Char	Meaning
`	The grave character represents a grave accent.
~	The tilde character represents a curved diacritical mark: a tilde or a breve.
^	The circumflex character represents a circumflex accent.
*	The asterisk character represents a circle-like mark: a ring above or a dot above. With lowercase i, it indicates the lack of a dot above; with e, it indicates a leading letter o.
'	The apostrophe character represents an acute accent, but is also used with s to represent ß.
''	The quote character represents a diaeresis or umlaut.
/	The slash character represents miscellaneous diacriticals and variations: cedillas, strokes, bars, eth, thorn, ligatures, and so forth.

These characters are “dead keys:” when first typed, there is no output. When a second character is typed, an accented character is produced if the combination is recognized. If the combination is not recognized, the two characters typed are produced. If the second character is a space, the original character is produced.

The various letters generated with each of the dead keys are shown in this table:

Char	Letters Produced
`	à è ì ò ù À È Ì Ò Ù
~	ã ģ ñ õ ù Ã Ğ Ñ Õ Ù
^	â ê ģ ĥ î ð ð û Â Ê Ĝ Ĥ Î ð ð û
*	â ç œ (with e) ĝ ï ž Ā Ć Œ Ġ Ī Ž
'	á é í ó ß (with s) ú ý Á É Í Ó Ú Ý
''	ä ë ï ö ü ÿ Ä Ë Ì Ö Ü Ÿ

Char	Letters Produced
/	ç ð (with d) æ (with e) ħ ø § þ (with y) Ç Ð Æ Ĥ Ø Ş Þ

Generally the key following the dead key corresponds to the basic Latin letterform without the diacritical mark: “Ĥ,” for example, is entered as “/H,” and “ğ” as “~g.” Where the relationship is not obvious (as with ß and æ), the appropriate letter key is noted. The letters eth and thorn are assigned to “d” and “y” respectively, based on their general appearance.

Symbol Characters

While dead keys are used for letters, alternate graphics are used for symbols. Symbols include spacing versions of the various accents. The alternate graphics provided are shown in this table:

Character	1	2	3	4	5	6
Alternate	ı	ı	ı	¼	½	¾
Character	a	c	o	s	u	x
Alternate	ª	¢	º	§	µ	×
Character	C	E	L	O	P	R
Alternate	©	€	£	°	¶	®
Character	Y	`	~	!	\$	*
Alternate	¥	·	˘	¡	¤	°
Character	()	<	>	-	_
Alternate	<	>	«	»	¬	-
Character	[]	{	}	+	
Alternate	'	'	“	”	±	‡
Character	:	“	.	/	?	'
Alternate	÷	“	.	,	¿	'

Note that the alternate for lowercase o is the masculine ordinal indicator, the alternate for uppercase O is the degree sign, and the alternate for asterisk is ring above. Although these characters may look similar, they are distinct. Similarly, the alternate for period is middle dot, while the alternate for grave is dot above.

Installing the Keyboard Layout

Installing the keyboard layout on a Windows machine is straightforward.

- The ZIP file containing the keyboard layout must be unpacked.
- An administrator must run the *setup.exe* program.

Selecting the keyboard layout for use is similarly straightforward. The following instructions are for Windows XP; the actions for Windows 2000 should be similar.

- The user clicks on Start | Control Panel | Regional and Language Options. This brings up the Regional and Language Options dialog.
- The user selects the Languages tab and clicks Details. This brings up the Text Services and Input Languages dialog.
- Under Installed Services, click Add. This brings up the Add Input language dialog. Under Input language, select "English (United States)." Under Keyboard layout/IME, select "US Layout with Latin-1 and Latin-3." This is the keyboard layout just installed.
- Click OK on the Add Input language dialog.
- Click OK on the Text Services and Input Languages dialog box.
- Click OK on the Regional and Languages Options dialog box.

The user should now have the language bar next to the system tray, and can use the keyboard icon to switch between keyboard layouts.

Invoking the Keyboard Layout

The keyboard layout may be used with any program. Select the keyboard layout using the language bar and start typing. Unicode aware programs will receive the Unicode characters specified by the key sequences you type.

Non-Unicode aware programs will be presented characters in the default code page for the language in use, which will in this case will be code page 1252. This code page includes all Latin-1 characters, but few Latin-3 characters. Many characters used in Western European languages will be available, but in general accented consonants will not be available.

Your program may ask you to perform a "save as" action when saving files using these characters. This may involve selecting an encoding. UTF-8 is the encoding typically used for Unicode data consisting primarily of Latin characters.

Glossary of Acronyms Used

ANSI: The American National Standards Institute.

ASCII: American Standard Code for Information Interchange.

ECMA: The European Computer Manufacturer's association.

IME: Input Method Editor.

ISO: The International Standards Organization.

UTF: Unicode Transformation Format.

Further Information

Jukka "Yucca" Korpela provides a highly detailed and informative look at the various issues involved in character codes at <http://www.cs.tut.fi/~jkorpela/chars.html>.

The Unicode Consortium provides definitive information on Unicode at <http://www.unicode.org/>.

ECMA Latin-1 through Latin-4 are defined in ECMA-094 available at <http://www.ecma-international.org/publications/standards/Ecma-094.htm>.

The Microsoft Keyboard Layout Creator is described in and can be downloaded at <http://www.microsoft.com/globaldev/tools/msklc.msp>.

This paper, the installable keyboard layout, and the source description for the keyboard layout are available at <http://www.brianhetrick.com/tr/AN-1.zip>.

Copyright and Grant of License

This Application Note is copyright © 2008 Brian Hetrick. It may be used and copied in accordance with the terms of the Creative Commons Attribution-Noncommercial-ShareAlike 3.0 United States license. The text of this license is available at: <http://creativecommons.org/licenses/by-nc-sa/3.0/us/>.

The software described herein is copyright © 2008 Brian Hetrick. It may be used and copied in accordance with the GNU General Public License, version 3. The

A Latin-1 and Latin-3 Characters US Keyboard Layout for Microsoft® Windows

text of this license is available at:
<http://www.gnu.org/licenses/gpl-3.0.html>.

This Application Note and the software it describes may be licensed for use and copying under terms oth-

er than those of the licenses described above, typically for fees of \$10 (US) per copy. Contact the author to make arrangements for such other licensing.

BrianHetrick.com