

Proposal for a new Keyboard Layout

2005-11-02

This is the 1st revision of the original publication dated 2005-10-31
(the recipients of which have received an errata)

Prepared for an open public review in Finland at the kotoistus web site and for comments by the relevant bodies in other Nordic countries and the international IT industry.

Source: the Keyboard Working Group reporting to the Finnish national Steering Group on Cultural Diversity Issues in ICT.

The proposed new general purpose keyboard layout extends the functionality of the presently common SWE/FIN keyboard layout considerably.

In spite of this, there are no disturbing changes to the users of the base functions of the regular (i.e., desk top) keyboards. In particular, there is no change to the allocation of the base letters. The number and placement of keys is the same as in the full size qwerty keyboards currently in use.

The proposed keyboard layout is designed to produce UCS/Unicode encoding. If a more limited encoding is required, the functionality of the keyboard will have to be restricted correspondingly.

The design is open-ended and based on intuitively recognizable positioning of additional letters and diacritical marks. It can thus be learned with a minimum effort on an as-needed basis.

User Communities

The proposed keyboard layout is intended to have a potential to be used by the speakers of the majority languages of Finland and Sweden, on the one hand, and Norway and possibly even Denmark, on the other hand. For the latter use, only two key positions (C10 and C11, as explained later) need to have obvious differences as compared with the Finnish/Swedish keyboard.

Basic Goals

The keyboard layout is designed to meet three important basic goals.

First, it provides for easy entering of text in the majority languages, i.e. Finnish, Swedish, Norwegian, and Danish, using a familiar keyboard layout.

Second, it is designed to give an easy, intuitively recognizable way to enter text in all of the official regional and minority languages, including those with the requirement for additional characters (i.e., for Finland, Sweden and Norway: Northern Sámi, Southern Sámi, Lule Sámi, Inari Sámi, Skolt Sámi, and Romany as spoken in Finland, and for Denmark: Faroese and Kalaallisut aka Greenlandic).

As a third objective, it allows for relatively easy entering of particularly the names (of persons, places or products) with many Latin characters used in other languages, such as the official languages of the European Union (though not, of course, including Greek) plus the remaining Nordic language, Icelandic.

The Latin characters in modern use of the MES-2 (Multilingual European Subset 2 of the UCS/Unicode) repertoire are the basis for this design. The needs of writing Vietnamese names have also been taken into account due to the size of that immigration population.

The Means

To achieve the goals, room is needed for non-decomposable characters in intuitively recognizable positions. For this purpose, almost all characters that are decomposable in UCS/Unicode terms are to be produced using a systematic approach utilizing dead keys to add diacritical marks. The letters å, ä, and ö are however kept in their currently allocated positions, for obvious reasons. (The letters ä and ö would switch places with their counterparts, æ and ø, in Norway and also their key positions in Denmark.)

It is also of particular importance to the end users that specific punctuation is provided to support the orthographies of the languages in question.

Presently, the basic ASCII quotation marks and hyphen-dash are often used by various applications (correctly or incorrectly) to produce localized punctuation. These characters are now augmented by new punctuation marks that produce the desired result independently of the application programs. With these additions, the keyboard supports much better the proper punctuation of the Nordic languages.

In addition, the no-break space (and possibly a narrow no-break space, if feasible) is provided as an easy to use tool for preventing unsuitable line breaks when texts are formatted.

Thus, there would be considerable justification to call the proposed keyboard layout “Nordic International”. It covers a superset of the repertoire covered by the current Swedish proposal (SIS remiss 4027) for a “Swedish International” layout, but employs a totally different design approach.

Compositions

The keyboard layout in current use has two specific (dead) keys for diacritics.

The proposal preserves these keys with some additional diacritics and one repositioning. The caron, used in Finnish and several of the regional and minority languages, is proposed to take the current place of the diaeresis. On the downside, the entering of the relatively common letter ü thus becomes less easy than presently.

In addition to the original dead keys, several keys gain dead key functionality when used in conjunction with the Alt Gr key (and possibly a Shift key). For example, the comma key acts as a dead key for the cedilla when used together with the Alt Gr key. In the sequel, the term “dead key” refers to both the two dead keys proper and the dead key functionality of other keys.

This design is most naturally implemented by defining that a dead key logically produces a UCS/Unicode combining diacritical mark after the character that is typed next. Thus, pressing the upper dead key (with the engravings ´ and `) and then a letter key produces that letter followed by combining acute accent (0301). However, the keyboard driver is expected to convert this to a precomposed character, such as á (00E1) or é (00E9), whenever possible, i.e. when such a character exists in Unicode version 3.0. This implies no change to current functionality for ISO Latin 1 characters, but it extends them in a well-defined, universal way. (The reference to Unicode 3.0 specifically is meant to make the definition stable, the same way as the Unicode normalization is stable.)

As a consequence, multiple, different diacritical marks are allowed on the same letter. Thus, e.g., diaeresis + caron + u would result in ů (01DA as a precomposed character). Such functionality is needed for correct writing of e.g. Vietnamese.

A special definition is suggested for some cases where a dead key is pressed twice in succession. Producing two identical combining diacritical marks would make little sense, so by convention, particular meanings for other, “double” diacritics are assigned.

– When the combining acute accent ´ (0301) dead key is pressed twice in succession, this corresponds to appending a different diacritical mark, the combining double acute accent ˝ (030B) to the next character. This diacritical mark would then be composed with its base character according to the same principle as any other diacritical marks. For example, pressing the acute accent key twice and then the u key would produce ű (0171), used in Hungarian. Similarly, pressing the combining grave accent ` (0300) dead key twice, could produce a combining double grave accent (030F). Several combining double diacritical marks are defined in UCS/Unicode and could be created in this fashion, but not all of them need be supported.

– As a different special case, using the dead key for combining dot below twice could correspond to a middle dot as follows. When followed by a space, it could produce a middle dot · as a separate character (00B7). When followed by a letter l or L, it could create ł (0140) or Ł (013F).

The composition principle is extended to some characters that are not composite in the UCS/Unicode sense but can intuitively be seen as consisting of a basic Latin letter and a stroke, such as đ. The stroke in such characters is not a diacritic, but in the proposed keyboard layout, a special modifier key is used to create a character with a stroke. Although the stroke is not a decomposable element of a character, a keyboard driver can handle it in a manner comparable to the use of dead keys for diacritical marks. This makes typing such letters easy and natural, as soon as the user has learned the simple principle. The proposed engraving of the key would suggest the principle and remind of it.

For ease of typing many Sámi letters and other characters with a stroke, the modifier function has been assigned as a primary function to a key. This inevitably means a change, but to minimize the problems, as well as to give the key an easily noticeable placement, the key E00 is used (when available). For this reason, the character § has been moved into the Alt Gr position. This implies a small inconvenience to typing legal texts.

This approach makes it possible to type several Sámi letters using just two keys, the modifier key and a basic letter key. The characters involved are đ - d with stroke (0111, in Northern, Inari and Skolt Sámi), ġ - g with stroke (01E5, in Skolt Sámi), and ɛ - t with stroke (0167, in Northern Sámi).

– The stroke is also used to create e.g. ł - l with stroke (0142, in Polish) or ħ - h with stroke (0127, in Maltese). Since the stroke is not a decomposable element, all combinations of the modifier key and a letter that produce a character can be specified separately. (The logically simplest approach would be to allow any Latin letter with the word “stroke“ in its UCS/Unicode name to be produced that way, but not all of them need be supported.)

When used in any other combination, the modifier key would have no effect. Alternatively, it could be defined to produce the character § in such cases. Thus, in practically all situations where the character § is needed, it could be typed as previously, though with a delay in its appearance on screen.

The Changes

The proposal means two changes to the currently used keyboard layout:

1. At E00, the character § has been moved into the Alt Gr position, in order to facilitate the easy entering of the stroke. (If E00 is not available on the keyboard, then a more laborous position at C09 would be used as a fallback).
2. At D12, the combining diaeresis has been moved to the Alt Gr + Shift position, in order to facilitate the easy entering of the combining caron.

The latter change is meant to facilitate specifically the easy entering of the Finnish letters š and ž and several other Sami and Finnish Romany letters: č - c with caron (010D, Northern, Inari and Skolt Sámi), ě - g with caron (01E7, Skolt Sámi), ě - h with caron (021F, Finnish Romany), ě - k with caron (01E9, Skolt Sámi), and ž - ezh with caron (01EF, Skolt Sámi). Among letters with a combining diaeresis, ä and ö are present on the keyboard of course, and ü is clearly the most common of the remaining letters in texts in Nordic languages, whereas ë and ï are rare and ÿ is extremely rare.

The proposal also changes the meaning of some key sequences, but this will probably not cause much difficulties. At present, pressing e.g. the acute dead key and the c key produces ´c, whereas in the proposed scheme, it produces ć. The sequence ´c would need to be produced by pressing the space key between the acute key and the c key. This is how users generally behave anyway even now.

The additional letters and diacritical marks are shown on the layout.

Open-endedness

The proposed layout is open ended. Only a minimum repertoire is defined.

In line with the current UCS/Unicode practice, the proposed keyboard driver does not check the applicability of letters with combining diacritical marks. Thus, the responsibility for checking the acceptability of any resulting character belongs initially to the user (by means of viewing the display of what has been input) and ultimately to the application (which may set its own limitations to the repertoire – as always, since e.g. names in a population registry may not contain otherwise perfectly valid numbers).

For the sake of consistent application of the design principle, no other decomposable letters than å, ä and ö are proposed to have their own key position. In practice, however, some such characters could be assigned without severely jeopardizing the open-endedness, if there is a demonstrated need.

A case in point could be the letter ü.

In Appendix 1, the specific new function (when compared with the common keyboard layout in current use) of each key is discussed.

Appendix 2 shows the proposed keyboard layout(s).

Appendix 1:

The functions of the various data keys are as follows:

<i>Pos</i>	<i>Base function</i>	<i>+ Shift</i>	<i>Alt Gr function</i>	<i>Alt Gr + Shift</i>	<i>Comment</i>
E00	Modifier: stroke	½	§		Repositioning + added modifier. If no E00, see C09!
E01	1	!			
E02	2	"	@	” (201D)	Added quotation mark.
E03	3	#	£	» (00BB)	Added quotation mark.
E04	4	α	\$	« (00AB)	Added quotation mark (for Norwegian and Danish).
E05	5	%		‰ (2030)	Added per mille sign.
E06	6	&			
E07	7	/	{		
E08	8	([
E09	9)]		
E10	0	=	}	° (00B0)	Added degree sign.
E11	+	?	\		
E12	Cd: ´ acute	Cd: ` grave	Cd: ´ horn (031B)	Cd: ´ hook above (0309)	Added diacritics (esp. for Vietnamese).
D01	q	Q			
D02	w	W			
D03	e	E	€		
D04	r	R			
D05	t	T	þ (00FE)	Þ (00DE)	Added letter thorn.
D06	y	Y			
D07	u	U			
D08	i	I	ı (0131)	İ (0130)	Added dotting variants for i.
D09	o	O	œ (0153)	Œ (0152)	Added oe ligature.
D10	p	P			
D11	å	Å	Cd: ° ring above (030A)	Cd: ˘ breve (0306)	Added diacritics.
D12	Cd: ˇ caron (030C)	Cd: ^ circumflex	Cd: ˜ tilde	Cd: ¨ diaeresis	Repositioning + added diacritic.
C01	a	A	ə (0259)	Ə (018F)	Added letter schwa.
C02	s	S	ß (00DF)		Added letter sharp s.
C03	d	D	ð (00F0)	Ð (00D0)	Added letter eth.
C04	f	F			
C05	g	G			
C06	h	H			
C07	j	J			
C08	k	K	κ (0138)		Added letter kra.
C09	l	L			If no E00 on the keyboard, fallback for stroke modifier at Alt Gr position.
C10	ö	Ö	ø (00F8)	Ø (00D8)	Added letter o with stroke.
C11	ä	Ä	æ (00E6)	Æ (00C6)	Added letter ae.

C12	'	*	' (2019)	Cd: , comma below (0326)	Added quotation mark + diacritic.
B01	<	>			
B02	z	Z	Ʒ (0292)	Ʒ (01B7)	Added letter ezh.
B03	x	X			
B04	c	C			
B05	v	V			
B06	b	B			
B07	n	N	ŋ (014B)	Ŋ (014A)	Added letter eng.
B08	m	M	μ (00B5)		Added micro sign.
B09	,	;	Cd: , cedilla (0327)	Cd: ˆ ogonek(0328)	Added diacritics.
B10	.	:	Cd: ˙ dot below (0323)	Cd: ˘ dot above (0307)	Added diacritics + special function for middle dot.
B11	-	–	– (2013)	Cd: ˉ macron (0304)	Added en dash + diacritic.
A	Space				Also no-break space (NBSP) and possibly a narrow NBSP; implementation tbd.

N.B. The layout intentionally leaves many positions unassigned, even though rather natural allocations could be presented. The need for additional assignments varies by the user, by the nature of the text, and by the program. Thus, they are best left unassigned in the basic layout, so that there is more room for varying useful customizations.