

Combinations of two diacritics

										Σ	
<b>a</b>	ā	á	(ā)					ǎ ǎ ǎ ǎ ǎ	ã ǎ ǎ ǎ ă		12
<b>c</b>	(ç)										–
<b>e</b>	(ë)				(é) (è)				ẽ ẽ ẽ é ê		5
<b>i</b>	(í)										–
<b>l</b>	(Ĺ)										–
<b>o</b>	ō	õ	ó	(ō)	(ó) (ò)	(ö) (õ) õ			õ õ õ ó ô	ỏ ỗ ờ ớ ợ	14
<b>n</b>	(ñ)										
<b>r</b>	(ř)										–
<b>u</b>	(ú)	(ü)					(ù) (ű)			ủ ữ ừ ú ự	5
<b>s</b>	(š)	(š)	(š)								–
										36	

Other letters plus diacritic

										Σ	
<b>æ</b>	æ	(æ)									1
<b>ž</b>	ž										1
<b>(dz)</b>	(dž)										–
										2	

2 BERND KAPPENBERG

	'	"	`	¨	^	ˆ	¨	¨	-	-	-	ˇ	˘	ˆ	ˇ	˘
<b>A a</b>	á		à	(ä)	â		ä		ā			ǎ		(â)	(ǎ)	
<b>B b</b>											(b)					
<b>C c</b>	ć				ĉ										č	ç
<b>D d</b>						(d)					(d)				ď	
<b>E e</b>	é		è	(è)	ê	(e)	ë		ē			ě		(ê)	ě	(e)
<b>F f</b>																
<b>G g</b>	(g)				ĝ				(g)			ğ			ğ	
<b>H h</b>					ĥ		ħ				(h)		(h)		ħ	(ħ)
<b>I i</b>	í		ì	(i)	î		ï		ī			ĩ		(î)	(i)	
<b>J j</b>					ĵ										(j)	
<b>K k</b>	(k)										(k)				ķ	
<b>L l</b>	ĺ					(l)					(l)				ļ	
<b>M m</b>	(m)															
<b>N n</b>	ń		ñ			(n)					(n)				ň	
<b>O o</b>	ó	õ	ò	(ö)	ô		ö		ō	(o)		ő		(ô)	(ö)	
<b>P p</b>	(p)															
<b>Q q</b>																
<b>R r</b>	ř			(ř)							(r)			(ř)	ř	
<b>S s f</b>	ś				ŝ										š	ş
<b>T t</b>						(t)	(t)				(t)				ť	(t)
<b>U u</b>	ú	ů	ù	(ü)	û	(u)	ü	(u)	ū			ů		(û)	(ü)	
<b>V v</b>																
<b>W w</b>	w		w		w		w									
<b>X x</b>							x									
<b>Y y</b>	ý		ÿ		ÿ		ÿ		ȳ							
<b>Z z</b>	ź				ż						(z)				ž	
<b>Σ</b>	<b>13</b>	<b>2</b>	<b>8</b>	<b>-</b>	<b>13</b>	<b>-</b>	<b>9</b>	<b>-</b>	<b>6</b>	<b>-</b>	<b>-</b>	<b>6</b>	<b>-</b>	<b>-</b>	<b>12</b>	<b>2</b>

,	·	·	.	/ /	-	ç	°	°	~	~	~	’	,				Σ
	à		ạ	(a)		ạ	å	(ạ)	ã			ả					12
	(b)		(b)		(b)										(b)		-
	ç			(ç)													5
đ	(đ)		(đ)		đ										(đ)		3
	è		ẹ	(è)		ẹ			ẽ		(è)	ẻ					12
	(f)																-
g	g			(g)	g												6
	(h)		h		h									(h)			5
	ì		ị	(ì)	ị				ĩ		(ì)	ỉ					11
					(j)												1
k			(k)	(k)	(k)									(k)			2
l		l	l	l	(l)					(l)					(l)		6
	(m)		(m)														-
n	n		(n)	(n)					ñ					(n)			6
	ò		ọ	ø		ø			õ			ỏ	ớ				14
	(p)				(p)												-
				(q)	(q)												-
r	(r)		(r)	(r)	(r)												3
s	(s f)		(s)	(s f)	(f)												5
t	(t)		(t)	(t)	t												3
			(u)			u	ù		(ũ)		(u)	ú	ư				11
			(v)	(v)					(ṽ)								-
	(w)		(w)				(ẇ)										4
	x																2
	(y)		(y)		(y)		(ẏ)		ÿ			ÿ					7
	z		(z)		(z)									(z)			4
8	9	1	6	2	4	5	2	-	6	-	-	6	2	-	-	-	122

(Letters in brackets means not used in European languages or Vietnamese.)

There are also so-called *Named Sequences*, sequences with combining diacritics that were not included as separate Unicode characters:<sup>1</sup>

Lithuanian dictionaries etc.	diverse*
ą ą̃	à
ė ė̃ é ě ě̃	ē ě
į į̃ í í̃	ì í
ĵ	
ĩ	
ĩ	
ĩ	
ĩ	
	ñg
ř	
ú ũ ú̃ ú̄	ù

\*apparently not used in  
European languages

Therefore, I have developed an own input method, which uses only ASCII characters:

<sup>1</sup> <http://unicode.org/Public/UNIDATA/NamedSequences.txt>  
(2012-02-11)

# 1 Development of a practical input method for characters

As the title of the input method I have chosen *Šibboleth*. Background is a passage from the Old Testament (Book of Judges, chapter 12). It reads:

» 5 The Gileadites captured the fords of the Jordan leading to Ephraim, and whenever a survivor of Ephraim said, “Let me cross over,” the men of Gilead asked him, “Are you an Ephraimite?” If he replied, “No,”  
6 they said, “All right, say ‘Šibboleth.’” If he said, “Sibboleth,” because he could not pronounce the word correctly, they seized him and killed him at the fords of the Jordan. «

## 1.1 Fundamental considerations

1. Existing input methods often work only under certain programs (e.g. MS Word) or only for certain subsets (e.g. Latin-1). There is therefore a need for a *universal* input method that can be used for all Latin characters of modern European languages (at least on MS Windows systems, better yet across platforms).
2. Input methods based on decimal and hexadecimal *numerical* values are not practical because the number of values corresponds by defini-

tion to the number of characters. The average user cannot be expected to retain about 400 numbers in memory or to always have a list of them handy.

3. The execution of all necessary diacritics as *dead keys* (like  $\hat{\ } \acute{\ } \grave{\ }$ ) would require the introduction of at least 14 additional key assignments. These new assignments would have to be *physically* marked on the keyboards. In the long term, existing international standards for keyboard layouts would have to be adjusted accordingly.
  - The Neo keyboard layout,<sup>2</sup> the *Europatastatur*<sup>3</sup> (“European keyboard”) and MEEK (*Functional Multilingual Extensions to European keyboard layouts*)<sup>4</sup> are following this approach. The problem is that they are all at least driver-based (if not hardware-based) solutions, i. e. they are not portable.
4. The execution of diacritics as dead keys was a technical necessity of the mechanical typewriter. In the age of electronic text processing, this historically concession is no longer mandatory.
5. Some systems use default escape codes based on existing key assignments. The Ä can be generated e. g. by means of  $\backslash \text{"A}$  (T<sub>E</sub>X source) or **Strg**  $\uparrow$  **:** **A** (MS Word).
6. The disadvantage of dead keys and default escape codes is that they are inconsistent with both the manual flow of writing as well as the nomenclature of characters (it is “A with diaeresis”, not “diaeresis over A”).

<sup>2</sup> [http://de.wikipedia.org/wiki/Neo\\_%28Tastaturbelegung%29](http://de.wikipedia.org/wiki/Neo_%28Tastaturbelegung%29) (2012-01-01)

<sup>3</sup> <http://www.europatastatur.de/> (2012-01-04)

<sup>4</sup> <ftp://ftp.cen.eu/CEN/Sectors/List/ICT/CWAs/CWA-16108-2010-MEEK.pdf> (2012-01-01)

7. Escape encodings outside the source require a compose function. MS Word, for example, uses the `Strg` key. The disadvantage is that the user must press up to three buttons simultaneously, e. g. `Strg` `⌘` `:` for a diaeresis.
8. An alternative to keyboard short-cuts is called a *hotstring*, i. e. a string that trigger keyboard events. The string `A:` would, for example, as hotstring automatically replaced by Ä. The problem is that the replacement would take place also at text positions where a colon was actually desired. Thus, the string must be “armed” by an additional key. This key should be a control button or alternatively a character that is not used in texts (or only rarely). On most keyboards this is the number sign `#` or the plus sign `+`.
9. There are two ways of *ordering* a hotstring: `#A:` or `A:#`. Considering the fact that letters can carry *multiple* diacritics (example Ǟ), the second variant `A:-#` is preferable because then the end of the escape coding is clear (while `#A:-` could mean both Ǟ and Ä-).
10. The escape codes should consist exclusively of *ASCII characters* (e. g. characters which are encoded by the vast number of ASCII-derived character sets by the same numerical value) for maximum compatibility and interoperability.
11. The escape codes should be *intuitive*, e. g. their shape should remind of the respective diacritic.

The intuitive combination would be “basic letter + diacritic replacement + compose”. They would work for most letters with diacritics and also for ligatures and digraphs.

However, diacritics which can be also set above *and* below would cause a problem. There is therefore a need for a second function key for diacritics that are to be put underneath. Intuitively the underscore `[_]` comes to mind. But in practice, problems occurred when the user assigns file names (and the underscore is used to replace a space). Therefore, we cannot waive the compose key: `[_][#]`.

Is there a need for a *third* function key, for letters that are *superimposed* by the diacritic? I would simply use the `[x]`. So:

1. base letter + diacritic replacement(s) + compose (= standard diacritic)
2. base letter + diacritic replacement + underneath + compose (= diacritic underneath)
3. base letter 1 + base letter 2 + compose (= letter combinations)
4. base letter 1 + base letter 2 + ligature/digraph signal `[j]` + compose (= only dz fi ffi)
5. base letter 1 + base letter 2 + ligature/digraph signal `[j]` + diacritic replacement + compose (= only ž dž æ é)
6. basic letter + diacritic replacement + overlay `[x]` + compose (= for - and ~)

### 1.1.1 Potential conflicts

Some cases of conflict must first be resolved:

The combination `[a][e][#]` could principally trigger ä, å, æ and ę. I have chosen the following combinations: `[a][:][#]` = ä, `[a][l][e][#]` or `[ä][e][#]` = å, `[a][e][#]` = æ and `[e][&][#]` = ę. These combinations would have a corresponding impact on the generation of œ/ô/ö and ù/ü



The Dutch ligature ij and the Croatian digraphs lj and nj are to be expected more frequent than the j (j without dot, probably only in medieval texts). Therefore  $\boxed{i}\boxed{j}\boxed{\#}$  = ij,  $\boxed{n}\boxed{j}\boxed{\#}$  = nj, etc. should get the privilege (= without ligature/digraph signal), and the j should be generated via  $\boxed{j}\boxed{0}\boxed{\#}$  instead of  $\boxed{j}\boxed{\#}$ .

Another case of conflict is the combination  $\boxed{d}\boxed{z}\boxed{\#}$  that could principally trigger 3 (ezh) as well as the digraph dz. A further complication is that both 3 and dz can also occur with a haček (ž, dž). If we set  $\boxed{d}\boxed{z}\boxed{\#}$  = 3, then dz must explicitly generated as digraph via  $\boxed{d}\boxed{z}\boxed{J}\boxed{\#}$ , even if this not coherent with lj and nj. It follows  $\boxed{d}\boxed{z}\boxed{J}\boxed{<}\boxed{\#}$  = ž and  $\boxed{d}\boxed{z}\boxed{<}\boxed{J}\boxed{\#}$  = dž. Both solutions are with five keystrokes comparatively type-intensive, but probably cannot be further optimized. The combination  $\boxed{d}\boxed{z}\boxed{h}\boxed{\#}$  as an alternative would collide with  $\boxed{z}\boxed{h}\boxed{\#}$  = ž.

In the same way,  $\boxed{a}\boxed{e}\boxed{\#}$  stands for æ but  $\boxed{a}\boxed{e}\boxed{J}\boxed{-}\boxed{\#}$  /  $\boxed{a}\boxed{e}\boxed{J}\boxed{'}\boxed{\#}$  for  $\bar{æ}$  /  $\acute{æ}$ .

The ligature fi is triggered the standard way  $\boxed{f}\boxed{i}\boxed{J}\boxed{\#}$  but ffi is triggered  $\boxed{F}\boxed{i}\boxed{J}\boxed{\#}$  (everything else did not work).

## 1.2 Escape encodings with diacritics

Taking into consideration escape codes of T<sub>E</sub>X and MS Word, I have chosen the following assignments:

acute ´	apostrophe ' □
double acute ¨	quotation mark " □
breve ˘	opening parenthesis ( □

inverted breve $\hat{\text{~}}$	closing parenthesis $)$
broad breve $\text{~}$	two times opening parenthesis $(($
broad inverted breve $\text{~}^{\frown}$	two times closing parenthesis $)$
breve underneath $\text{~}_$	closing parenthesis + underscore $)_$
inverted breve underneath $\text{~}_$	opening parenthesis + underscore $(_$
cedilla $¸$	semicolon $;$
descender	digit $4$
grave $`$	backslash $\backslash$
double grave $``$	two times backslash $\backslash\backslash$
háček $ˇ$	less-than sign $<$
háček underneath $ˇ_$	less-than sign + underline $<_$
hook $ˆ$	question mark $?$
hook above/underneath	(experimental:) section sign $§$ [no ASCII character]
horn $ˆ$	digit $9$
comma $,$	comma $,$
macron $ˉ$	hyphen/minus $-$
macron in centre	hyphen/minus + x $-x$
macron underneath $_$	hyphen/minus + underscore $-_$
double macron underneath $__$	two times hyphen/minus + underscore $--_$
ogonek $˛$	& sign $&$
dot $˙$ / dot in centre $\cdot$	dot $.$

dot underneath . (alternative) <sup>5</sup>	exclamation mark [!]
dash -	equal sign [=]
ring °	asterisk [*] / degree symbol [°] [no ASCII character]
slash /	slash [/]
tilde ~	tilde [~]
tilde in centre	tilde + x [~][x]
tilde underneath ~	tilde + underscore [~][_]
broad tilde ~	two times tilde [~][~]
trema ¨	colon [:]
top-bar ]	closing square bracket [)]
circumflex ^	greater-than sign [>]
circumflex underneath ^	greater-than sign + underscore [>][_]
with long leg	vertical line [I]
inverted letter	vertical line [I]
stand-alone diacritic	space + diacritic
combining diacritic	opening square bracket [I] + diacritic

The operation of the program is very easy: enter the basic letter(s), then the diacritic(s), then the number/underscore key.

Examples: [n]['][#] =  $\acute{n}$  [u]["][#] =  $\acute{u}$  [g][()][#] =  $\check{g}$  [e][)][#] =  $\hat{e}$   
 [i][)][)][#][a] =  $\hat{i}\hat{a}$  [c][;][#] =  $\csc$  [n][4][#] =  $n_4$  [y][\][#] =  $\grave{y}$  [r][\][\][#] =  $\ddot{r}$

<sup>5</sup> especially where also one diacritic above exists

`c < #` = č   `a ? #` = á   `z § #` = ž   `o 9 #` = ó   `t , #` = †  
`u - #` = ū   `o - x #` = ø   `e & #` = ě   `e . #` = è   `s ! #` = š  
`z = #` = z   `u ° #` = ů   `o / #` = ø   `a ~ #` = ã   `l ~ x #` = ł  
`u ~ _` = ŭ   `x : #` = ẍ   `b ] #` = ß   `y > #` = ŷ   `d > _` = đ

If the basic letter has several diacritics, first the diacritics under the letter and then the diacritics over the letter are entered (ascending).

Examples: `e ! > #` = ê   `u : < #` = ü

### 1.3 Escape encodings *without* diacritics

Some letters are entered directly (without diacritics):

Æ	<code>A E</code> AE ligature
IJ	<code>I J</code> IJ ligature
Œ	<code>O E</code> OE ligature
Ð	<code>D</code> Eth
Þ	<code>T H</code> Thorn / Þ <code>T H ] =</code> Thorn with dash = “that”
İ	<code>I</code> Turkish I
Ñ	<code>N N</code> N with tilde (additional)
ß	<code>S S</code> sharp S
Ŋ	<code>N G</code> Eng
Ʒ	<code>D Z</code> Ezh / Ʒ <code>D Z ] &lt;</code>
Ə	<code>Ä</code> Schwa (additional) [no ASCII character], <code>A : ]</code>
DZ	<code>D Z ]</code> DZ digraph / DŽ <code>D Z &lt; ]</code> DŽ digraph
LJ	<code>L J</code> LJ digraph

NJ	<b>N</b> <b>J</b> NJ digraph
Q	<b>K</b> <b>V</b> (additional)
W	<b>V</b> <b>V</b> (additional)
X	<b>K</b> <b>S</b> (additional)
Å	<b>A</b> <b>A</b> (additional)
Θ	<b>T</b> <b>0</b> Greek Theta (for Romani)
f	<b>s</b> <b> </b> long s / <b>s</b> <b> </b> <b>t</b> <b> </b> ft ligature
Ä	<b>Ä</b> <b>e</b> over-set e set in early forms of Ä (additional) [no ASCII character]
ƿ	<b>W</b> Wynn
ȝ	<b>G</b> <b>H</b> Yogh
f	<b>f</b> florin (guilder)
κ	<b>k</b> kra – The capital letter is K <sup>ˆ</sup> , i.e. K + singular quotation marks above.
Q̄	<b>Q</b> Gha
Θ̄	<b>Ö</b> (additional) [no ASCII character]
H̄	<b>H</b> <b>V</b> Hwair, for transcription of the letter Θ from the Gothic script
ŋ	<b>N</b> <b> </b> N with extended right leg
R	<b>Y</b> <b>R</b> for transcription of the Norse rune <i>elhaz</i> ᚱ
8	<b>O</b> <b>U</b> OU ligature, in Indian languages
F	<b>/</b> <b>H</b> claudianic <i>Sonus medius</i>

Ɔ	<b>B S</b> , <b>P S</b> claudianic <i>Antisigma</i>
I	<b>I / I</b> , <b>I I</b> , <b>I I</b> Latin <i>I longa</i> (aus Í)
Λ	<b>M M</b> Latin archaic M
J	<b>j o</b> j without dot
‡	<b>I = =</b> alveolar click
!	<b>! I</b> retroflex click
IL	<b>L L I</b> middle-Welsh LL
AA	<b>A A I</b> AA digraph
AO	<b>A O I</b> AO digraph
AJ	<b>A U I</b> AU digraph
AV	<b>A V I</b> AV digraph / <b>A V I =</b> AV digraph with dash
AY	<b>A Y I</b> AY digraph
OO	<b>O O I</b> OO digraph
W	<b>V Y I</b> VY digraph

For the other characters I have chosen the following assignments:

‰	<b>%</b> promille
@	<b>a t</b> at sign
¤	<b>i c s</b> international currency symbol
0 <sup>a</sup> ... 9 <sup>a</sup>	<b>0 a</b> ... <b>9 a</b> numbers, female
0 <sup>o</sup> ... 9 <sup>o</sup>	<b>0 o</b> ... <b>9 o</b> numbers, male
¬	<b>N O T</b> NOT sign

°	<b>d e g</b> degree symbol
·	<b>l l .</b> stand-alone dot in the middle (for L· in Catalan)
¼	space <b>1 / 4</b>
	½ – space <b>1 / 2</b>
	¾ – space <b>3 / 4</b>
	⅓ – space <b>1 / 3</b>
	⅔ – space <b>2 / 3</b>
	⅕ – space <b>1 / 5</b>
	⅖ – space <b>2 / 5</b>
	⅜ – space <b>3 / 5</b>
	⅘ – space <b>4 / 5</b>
	⅙ – space <b>1 / 6</b>
	⅚ – space <b>5 / 6</b>
	⅛ – space <b>1 / 8</b>
	⅜ – space <b>3 / 8</b>
	⅝ – space <b>5 / 8</b>
	⅞ – space <b>7 / 8</b>
	1/- space <b>1 /</b>
”	space <b>' ' space</b> – inch
‡	<b>? !</b> interrobang
‰	<b>a / c</b> account of ...
‰ <sub>s</sub>	<b>a / s</b> addressed to the subject ...
‰	<b>c / o</b> care of ...
‰ <sub>u</sub>	<b>c / u</b> <i>cada una</i> (each piece)
°C	<b>° C</b> , <b>d C</b> degrees Celsius
°F	<b>° F</b> , <b>d F</b> degrees Fahrenheit

™	<b>T</b> <b>M</b>	trademark sign
☺	<b>:</b> <b>)</b>	Smiley
☹	<b>:</b> <b>(</b>	Frowny
Å	<b>A</b> <b>/</b> <b>S</b>	<i>Aktieselskab</i> (Danish for company)
	space <b> </b>	broken bar
→	<b>-</b> <b>-</b> <b>&gt;</b>	arrow
➔	<b>-</b> <b>/</b> <b>-</b> <b>&gt;</b>	arrow struck through
↔	<b>&lt;</b> <b>-</b> <b>&gt;</b>	arrow with two heads
⇒	<b>=</b> <b>=</b> <b>&gt;</b>	thick arrow
⇔	<b>&lt;</b> <b>=</b> <b>&gt;</b>	thick arrow with two heads
Σ	<b>s</b> <b>u</b> <b>m</b>	sum sign
№	<b>N</b> <b>o</b>	number sign
⊕	<b>(</b> <b>+</b> <b>)</b>	circled plus sign
⊖	<b>(</b> <b>-</b> <b>)</b>	circled minus sign
∅	<b>o</b> <b>/</b>	diameter
ϣ	<b>x</b> <b>p</b>	chi-rho
卐	<b>x</b> <b>x</b>	right-wing Swastika
☘	<b>*</b> <b>)</b>	hammer and sickle
☾	<b>(</b> <b>*</b>	crescent and star
☯	<b>Y</b> <b>Y</b>	yin-yang

|| Do not forget to add **#**! ||



## 1.4 Underscore

With only an underscore  $\_$ , the following characters are made:

¡  $\! \_$  Spanish inverted exclamation mark

¿  $\? \_$  Spanish inverted question mark

Ƶ  $\! \! \_$

|  $\! \_$  dental click

||  $\! \! \_$  lateral click

With underscore  $\_$  plus compose key  $\#$ , the following characters are made:

Ɔ  $\! \_$  African O

Ɔ  $\! \_$  claudian *Digamma inversum*

and other reverse letters ( $\! \_$ , etc.)

∞  $\! \_$  infinity symbol

## 1.5 Other characters (no compose key)

„“ space  $\" \"$  ...  $\" \"$  space – exclamation marks (default German)

» « space  $\> \>$  ...  $\< \<$  space – *guillemets*

– space  $\- \-$  space – n-dash

— space  $\- \- \-$  space – m-dash

±  $\+ \-$  space – plus or minus

–  $\- \-$  real minus sign (on numeric keypad)

× space  \*  real multiplication sign

÷ space  /  real division sign

≥  >  = greater than or equal to

≤  <  = less than or equal to

≠  =  / not equal

≐  =  > corresponds to

≈  =  ~ almost equal

⇌  <  > chemical equilibrium

†  +  | dagger

‡  +  + double dagger

…  .  .  . ellipse

♥  <  3 heart

7  &  & Tironian et

Currency symbols: \$ + ISO-4217 code (if there is a currency symbol)

₴  \$  u  a  h Ukrainian Hryvnia

## 1.6 Dead keys

The scope of the dead keys was extended. Now it is possible to enter all the letters with ´ ` , ¨ directly (also in combinations).

I used the scripting language *AutoHotkey\_L* for MS Window<sup>6</sup> to realize the above presented escape codes (and some others). The result is the commercial program *Šibboleth*.<sup>7</sup>

*IronAHK*, a rewritten *AutoHotkey* for .NET and Mono for the purpose of cross-platform application (Unix, Mac) is currently not yet finished.<sup>8</sup>

---

6 free of charge: <http://l.autohotkey.net/> (03.01.2012)

7 [www.sonderzeichenhilfe.de](http://www.sonderzeichenhilfe.de)

8 <http://www.autohotkey.com/board/topic/50354-ironahk-alpha-cross-platform-net-rewrite-of-autohotkey/page-39>