

Typing diacritics in JuniusX

JuniusX has better support for diacritics than any other free medieval font. You can place a diacritic (or “combining mark”) on just about any character where it makes any sense at all to place it. Just type the base character followed by the diacritic.

But typing a diacritic can be a problem. (I’m not talking about the common letter+diacritic combinations like á, for which your system no doubt provides special keyboard sequences, but about exotic combinations like þ̊ or m̊. Stuff that comes up with distressing regularity when you’re transcribing medieval manuscripts, whose scribes weren’t constrained by the limitations of mechanical typesetting.) Diacritics—the exotic ones, at least—have no assigned keys on a keyboard. Instead you need to know their *encodings*: the four-digit hexadecimal (base-16) number assigned to them in the Unicode standard.

If what I’ve just written looks like gibberish, don’t worry: JuniusX offers you a way around the pain of having to learn dozens of codes like “1DE1.” Specifically, it provides a collection of mnemonics for typing in diacritics. In some cases, these mnemonics are conveniences, while in other cases, using them will help you avoid a serious technical difficulty.

The mnemonics are all enclosed in braces {}, and they consist of the circumflex character (above the 6 on American keyboards) followed by one to four letters. In the simplest case, the sequence {^a} produces a combining a, which can go above any letter, e.g. m̊. Some diacritics are like small capitals, and you type those in the same way, but with “sc” after the letter. For example, the sequence t{^dsc} gives you t̊—a t with a small cap D above. Some are ligatures, and you type both elements of those, e.g. m{^an} = m̊. For some ligatures, the second element is a small cap, so Z{^ansc} = Z̊.

There are also mnemonics for diacritics that are not alphabetic characters. For example, {^zz} gives you a zigzag, that maddeningly ambiguous abbreviation for er, is, es and dozens of other things: æ̊.

There’s not much more to it, except for a few important caveats. First, these codes affect the *appearance* of your document but *not the underlying text*. When you type the sequence p{^aopn}, it looks like an open a on your screen (þ̊), but underneath, the text is still p{^aopn}. So if you send your text file to a friend who doesn’t have JuniusX installed, that friend will see the mnemonics, not the diacritics; if you copy your text onto the clipboard

and paste it into an application that either can't access JuniusX or can't apply OpenType features, you'll also see the mnemonics.

And so you may prefer typing those incomprehensible Unicode values to typing mnemonics, at least for the Unicode-standard diacritics. Indeed, if you are sending your text to a publisher, you really ought to do so, or perhaps perform a last-minute search-and-replace to replace the mnemonics with their Unicode equivalents.

The second caveat is that, while most of the diacritics in JuniusX belong to the Unicode standard, and any competent application will render them correctly, positioning them precisely over their base characters, many diacritics *do not* belong to Unicode, and some applications will not position them correctly: they may come out looking like “m̂,” or worse yet “Ā.” (You can tell these non-Unicode diacritics by the codes that start with “E” or “F,” e.g. “F03B”—their entries are in green in the table below.)

✍ But (this is important) if you use the mnemonics for these non-standard diacritics, they will always be positioned perfectly over their base characters. (If you really need to know why, brace yourself for a very technical lecture lasting about a half hour.)

A third caveat: in most text-processing applications, the features needed to make mnemonics work are on by default, *but in Microsoft Word they are not*. You've got to turn them on. To do so, open the “Font” dialog, click over to the “Advanced” tab, and enable **Kerning**, **Standard Ligatures**, and **Contextual Alternates**. Come to think of it, you should do this for every font you use (do it for the “Normal” style and check “Apply to Template” before clicking “OK”). You may be surprised what wonders a font is capable of performing when these three features are enabled.

Here are the mnemonics. I should mention that there are many more diacritics in JuniusX than there are mnemonics.

Encoding	Mnemonic	Example	Encoding	Mnemonic	Example
035B	{^zz}	ǃ	F03D	{^p}	ᵖ
1DD3	{^aopn}	ᵒ	1DE3	{^₂}	ᵒ
0305	{^ol}	ō	1DE5	{^f}	ᶠ
0363	{^a}	ᵃ	1DD8	{^δ}	ᵒ

Encoding	Mnemonic	Example	Encoding	Mnemonic	Example
F012	{^b}	^b ○	1DD5	{^æ}/{^ao}	^æ ○
0368	{^c}	^c ○	1DD6	{^a}/{^av}	^a ○
0369	{^d}	^d ○	F135	{^e}	^e ○
0364	{^e}	^e ○	F136	{^ē}	^ē ○
F017	{^f}	^f ○	F02F	{^i}	ⁱ ○
1DDA	{^g}	^g ○	F031	{^j}	^j ○
036A	{^h}	^h ○	F13E	{^o}	^o ○
0365	{^i}	ⁱ ○	F032	{^ø}	^ø ○
F030	{^j}	^j ○	F13F	{^ō}	^ō ○
1DD3	{^k}	^k ○	1DD2	{^us}	^u ○
1DDD	{^l}	^l ○	1DD1	{^ur}	^u ○
036B	{^m}	^m ○	F013	{^bsc}	^b ○
1DE0	{^n}	ⁿ ○	F016	{^dsc}	^d ○
0366	{^o}	^o ○	1DDB	{^gsc}	^g ○
F025	{^p}	^p ○	F01C	{^ksc}	^k ○
F033	{^q}	^q ○	1DDD	{^lsc}	^l ○
036C	{^r}	^r ○	036B	{^msc}	^m ○
1DE4	{^s}	^s ○	1DE0	{^nsc}	ⁿ ○
036D	{^t}	^t ○	036C	{^rsc}	^r ○
0367	{^u}	^u ○	F02A	{^tsc}	^t ○
036E	{^v}	^v ○	F036	{^an}	^a ○
F03C	{^w}	^w ○	F038	{^ar}	^a ○
036F	{^x}	^x ○	F03A	{^ansc}	^a ○
F02B	{^y}	^y ○	F130	{^arsc}	^a ○
1DE6	{^z}	^z ○	F03E	{^o ₂ }/{^or}	^{o₂} ○
1DD4	{^æ}	^æ ○	F03F	{^o ₂ }/{^orr}	^{o₂} ○
1DD9	{^δ}	^δ ○	F03B	{^τ}	^τ ○

Finally, JuniusX has five OpenType features for producing alternative shapes of certain diacritics. These are “Character Variants,” cv61–cv65:

cv42 transforms U+035B or {^zz} (̇) into (1) ̇ or (2) ̇. Alternatives: directly enter U+F1C7 or U+F1C8.

cv43 transforms U+1DD3 or {^aopn} (̈) into ̈. Alternative: U+F1C1.

cv44 transforms U+1DE3 or {^2} (̈) into ̈. Alternative: U+F1C2.

cv45 transforms U+0305 (̄) into (̄). Alternative: U+F1C0.

cv46 transforms U+0303 (̂) into (̂). Alternative: U+F1CC.

Unfortunately, MS Word provides no access to Character Variants, so you must directly enter non-standard character codes listed as alternatives above, though these will not always be positioned correctly over base characters.