

Universal Multiple-Octet Coded Character Set
International Organization for Standardization
Organisation Internationale de Normalisation
Международная организация по стандартизации

Doc Type: Working Group Document
Title: On the apostrophe and quotation mark, with a note on Egyptian transliteration characters
Source: Michael Everson
Status: Expert Contribution
Date: 1999-07-24

Recently discussion of the use and semantics of U+0027 APOSTROPHE, U+02BC MODIFIER LETTER APOSTROPHE, and U+2018 RIGHT SINGLE QUOTATION MARK arose on the unicode@unicode.org list. In this paper I endeavour to discuss the situation comprehensively. The document presents the glyphs of the characters in question at twice their normal size for clarity.

In traditional typography, there was never a problem. An apostrophe was always represented by ’. This mark was also used, singly and doubly, to represent quotation marks in some countries: ‘quotation’, “quotation”. Other traditions also arose, some using low quotation marks: ,quotation‘, „quotation“; and others using unidirectional quotation marks: ’quotation’, ”quotation”. The use of guillemets (literally ‘billies’ for ‘little Guillaumes’) »quotation«, or «quotation», is irrelevant to this discussion.

No matter what the tradition, typesetters working with metal type routinely set the requisite quotation marks in the appropriate places. The single apostrophe, a mark of elision (*Mark’s* < Middle English *Markes*; *l’apostrophe* < *le apostrophe*), always had the ’ shape.

The Hebrew letters ALEF and AYIN and the Arabic letters HAMZA and AIN, having no equivalents in the Latin alphabet, became represented in transliteration by apostrophe-like glyphs of various kinds. This practice goes back a long way. Palmer 1901 gives ’ for HAMZA and ‘ for AIN. Clarke 1878 gives , (!) for HAMZA and ’ for AIN. Faulmann 1880 uses a standard alphabet for transcribing all the scripts in his book; he settled on ’ for ALEF and } for AYIN in Hebrew, and ’ for ALEF and } for AIN in Arabic and Persian. Faulmann writes HAMZA with h‘.

(NOTE: The double-bellied AYIN is, I believe, missing from the UCS. It is used in the Egyptologist tradition (to represent the sound written ) and generally has the letterform ʾ (Collier & Manley 1998) or ʾ (Gardiner 1966); Loprieno 1995 gives the unusual ʾ (which could be U+025C) but his typography is atypical. Likewise, Egyptologists use a special *i* which has some kind of diacritic over it. This is problematic because, although the character only has one function (to represent the sound written ) it has several glyph forms in printed text. This character’s usual shape is *i* (Collier & Manley 1998) or *ī* (Gardiner 1966); Loprieno 1995 substitutes *j* (U+006A LATIN SMALL LETTER J). Betrò 1995 uses *î*. What diacritic is this? U+0313 COMBINING COMMA ABOVE? *COMBINING RIGHT HALF RING ABOVE? *COMBINING RIGHT ARROWHEAD ABOVE? Egyptologists also use a special modifier letter (to represent the sound written ) written *ʿ* (Collier & Manley 1998) or *ʿ* (Gardiner 1966); Loprieno 1995 gives only ‘ (U+02BB MODIFIER LETTER TURNED COMMA), but again, his

typography is atypical. The shape (height, rotation, length) and size of this in printed texts is quite different from U+02BD MODIFIER LETTER REVERSED COMMA ‘. I suggest the addition of 3 LATIN LETTER EGYPTOLOGICAL ALEF, non-decomposable *i* LATIN LETTER EGYPTOLOGICAL YOD, and ˆ LATIN LETTER EGYPTOLOGICAL AYIN to the UCS.)

In addition to the use of ‘ and ’ for Semitic transcription, the use of Greek smooth breathing ˆ and rough breathing ˆ were commonly known (and were possibly the sources for the Semitic transcription characters). For ALEF sometimes ˆ was used. At some point, special characters ˆ (U+02C1 MODIFIER LETTER LEFT HALF RING) and ˆ (U+02C1 MODIFIER LETTER RIGHT HALF RING) were devised. Linguists in various countries extended the use of these characters in various ways, to represent glottal consonants (true letters), aspiration or glottalization of base consonants (modifier letters), and so on. When unavailable in fonts, horrible substitutions like superscript ^C have been made. All this happened in the last 150 years or so.

The invention of the manual typewriter introduced a new complexity to the situation. It was considered impractical to “waste” two keys for the four symbols “ ” ‘ ’, since typewriters were used for business purposes, and characters like ½ and @ were considered more useful (this despite an admirable attempt ca. 1881 to introduce a typewriter with a mahogany piano-finish case and ebony keys to Victorian drawing-rooms (see <http://users.erols.com/chuck101/hammondscan.JPG>)). Thus "dumb quotes" and the single APOSTROPHE ' were born. This proved no great hardship, since metal typesetters set text in the same way from typescript as they had always done from manuscript. A photograph of the 1903 Kanzler German typewriter appears to have four quotation marks: „ ” , ’ (the , doubling as COMMA).

Additional problems arose with the advent of computer technology. The small number of graphic characters available in 7 bits (95) was close to the number of characters available on typewriters (85), and so the ASCII repertoire is broadly based on US and UK typewriter repertoires, with the addition of GRAVE, CIRCUMFLEX, as well as DIGIT ZERO, LATIN SMALL LETTER L, GREATER-THAN, LESS-THAN, REVERSE SOLIDUS, LEFT CURLY BRACKET, RIGHT CURLY BRACKET, and TILDE. Backspacing technology with early printers could (I presume) make use of APOSTROPHE, GRAVE, CIRCUMFLEX, and QUOTATION MARK to produce letters with ACUTE, GRAVE, CIRCUMFLEX, and DIAERESIS, but this soon proved inadequate and so national variants of ASCII were developed.

Fonts for VDU terminals were often designed having an acute-angled APOSTROPHE, and users often employed the GRAVE ACCENT to pair with it to represent what many of them considered typographically acceptable single quotes: `quotation´. However, double quotation marks were still incorrect; perhaps some users typed ``quotation´´, but I doubt whether this practice was widespread, since by this time many people had become used to "dumb quotes", after decades of typewriter use. But the practice of using these characters persists to the present day.

Even when 8-bit technology (ISO 8859) was introduced, space in Latin 1 was still at a premium, and the quotation marks “ ” ‘ ’, EN DASH –, and EM DASH — were not supported. Apple Computer, abandoning the x80–x9F restriction, introduced these characters in its Macintosh Roman character set, and subsequently Microsoft did the same in its extension of Latin 1.

I assume that in 7 bits the ALEF/AIN distinction would have been made with GRAVE ACCENT and APOSTROPHE, but ASCII and Latin 1 would not satisfy the needs of Semiticists in any case, as letters like *ħ* and *ḥ* are required. Most likely the required diacritics were often written in by hand whether typewriter technology or early computer technology was

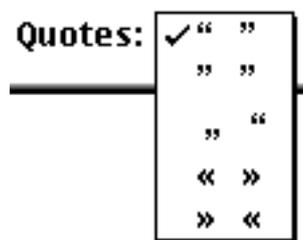
used, prior to being sent for typesetting.

This brings us to the present day, where the Universal Character Set contains all of these characters as individual units. There are two contexts which affect implementors: inputting of UCS data and conversion of non-UCS data to UCS format. Both of these contexts concern us, but in different ways.

Inputting requires a simple mechanism for the users to enter the characters they require. Many national keyboard layouts do not offer a simple means even to enter smart quotes and en and em dashes. The US/UK Macintosh keyboards, however, did provide the user with simple and easily-remembered inputting strategies. HYPHEN - is entered by typing HYPHEN; UNDERSCORE _ is SHIFT-HYPHEN; EN DASH – is OPTION-HYPHEN; EM DASH — is OPTION-SHIFT-HYPHEN. APOSTROPHE ' is found on APOSTROPHE, and QUOTATION MARK " on SHIFT-APOSTROPHE; LEFT SINGLE QUOTATION MARK ‘ is OPTION-]; RIGHT SINGLE QUOTATION MARK ’ is OPTION-SHIFT-]; LEFT DOUBLE QUOTATION MARK “ is OPTION-[; RIGHT DOUBLE QUOTATION MARK ” is OPTION-SHIFT-[. A number of wordprocessing and typesetting programs allow the user to set preferences for automatic presentation of contextual quotation marks when the basic APOSTROPHE and QUOTATION MARK keys are pressed. However, the algorithms used are *not* uniform. This is bad and confusing for users.

Nesting behaviour differs from program to program. Quark XPress requires the user to alternate single and double quotation marks in order to continue a run of opening quotes: “““““““. When the user types two of the same marks in a row, the second takes the closing form: ““““““”. In Nisus Writer, an unlimited number of opening quotation marks, alternating or not, is permitted as the closing marks are only invoked after a different character is entered: “““““““s”. In both ClarisWorks and Microsoft Word 98, correct nesting is not supported, because only the first quotation mark in a string is permitted to be an opening quote (inadequate for text presenting reported speech).

Quark XPress is also very powerful in that international quotation mark options are offered. Microsoft Word 98, Nisus Writer, and ClarisWorks do not offer these. Possibly they are hard-coded (specified in a localizable string resource) in localized versions.



Where all these algorithms fail is in their treatment of the punctuation apostrophe. Although it is uncommon in standard English, in a number of languages, such as Dutch, Irish Gaelic, and Scottish Gaelic, strings often occur with an initial punctuation apostrophe: 's-Gravenhage, 't Zand; 'S é do bheatha; 'Sann an-diugh a rinn mi sin. (In English it is probably known to most people only in phrases like “salt 'n' pepper”.) Since the punctuation apostrophe and the closing single quote both make use of U+2019 (or the Mac and Windows equivalents), automatic smart quotes produce the incorrect ‘s, ‘t, and ‘S in these contexts. *All general-use keyboards should provide a mechanism such as that described for the Macintosh above to allow the user force the correct apostrophical form in word-initial position.* No matter what national keyboard is selected by the user, a Dutch or Irish phrase might be required, and the user should have easy access to the

correct character. Even in monolingual texts, a discussion of quotation marks could occur, and both “” and “’” might be required.

A number of natural languages possessing a glottal stop choose, conventionally, to use either ‘ or ’ to represent it. Hawai‘ian uses the former; Azerbaijani, Navajo, and a number of languages using the Cyrillic alphabet use the latter. In all of these contexts, it is a UCS MODIFIER LETTER which is required. However, large amounts of data encoded for these languages using official or variant Macintosh or Windows character sets have (due to a lack of any other choice) used the punctuation apostrophe equivalents of U+2018 ‘ or U+2019 ’ for these. Conversion of this non-UCS data will be dealt with below. Nonetheless, it is clear that a simple mechanism for inputting these characters is required. Consider the following example, which contains all four characters: U+02BC, U+02BD, U+2018, and U+2019.

“Isn’t ‘Hawai‘ian’ the correct form and ‘Hawai’ian’ the incorrect form?” she asked.

Obviously along with these inputting options must go a certain amount of knowledge on the part of the users, who must learn that there *is* a correct character to enter in a given circumstance. Perhaps a paper like this one, suitably edited, should appear as a UTR or something.

I was interested to learn that UTR8 not only corrected the text on APOSTROPHE and RIGHT SINGLE QUOTATION MARK but also removed MODIFIER LETTER APOSTROPHE from the list of alphabetic characters. Perhaps this was a mistake. While this character is used as a modifier letter in the IPA (it indicates ejective (glottalic egressive) consonants like [p’ t’ k’]), it is in many natural languages a true letter of the alphabet, endowed with a place (usually at the end) in the alphabetic order. Examples from Mycaeb 1965: Azerbaijani, Nenets, Nivkh (as letter); Chukot, Eskimo, Khanty, Koryak, Kurdish, Selkup (as modifier letter).

If alphabetic property affects word selection (double-clicking to select an entire word) then perhaps the alphabetic property should be restored to U+02BC. (The same would be true for Nenets ’ U+20EE.) Ken Whistler pointed out in an e-mail to the unicode@unicode.org list that, in Word 97, strings containing punctuation apostrophe are correctly selected but that, while double quotation marks are correctly ignored in such selection, a closing single quote is mistakenly selected. I tested these environments with an Irish Gaelic word in the four programs and obtained the following results.

Quark XPress	Nisus Writer	ClarisWorks	Word 98
b’fhéidir	b’fhéidir	b’fhéidir	b’fhéidir
“b’fhéidir”	“b’fhéidir”	“b’fhéidir”	“b’fhéidir”
‘b’fhéidir’	‘b’fhéidir’	‘b’fhéidir’	‘b’fhéidir’

Quark selects everything between two spaces, and so does not take any character properties into account. Nisus Writer and ClarisWorks perform the selection correctly, and Word 98 has the same error as Ken reported for Word 97. When punctuation apostrophe appears string initially, however, none of the applications select correctly. Word almost succeeds, but breaks at a hyphen, which is not expected. Or at least *I* wouldn’t expect it. Quark selects correctly only in the first instance, but this is accidental.

Quark XPress	Nisus Writer	ClarisWorks	Word 98...	...Word 98
's-Gravenhage	's-Gravenhage	's-Gravenhage	's-Gravenhage	's-Gravenhage
‘s-Gravenhage’	“s-Gravenhage”	‘s-Gravenhage’	‘s-Gravenhage’	‘s-Gravenhage’
“s-Gravenhage”	“s-Gravenhage”	“s-Gravenhage”	“s-Gravenhage”	“s-Gravenhage”

Conversion of non-UCS data to UCS data which can be correctly managed by search engines, spell checkers, or sorting algorithms is of particular interest in the context of sixteen characters:

- U+0022 QUOTATION MARK "
- U+0027 APOSTROPHE '
- U+0060 GRAVE ACCENT `
- U+00B4 ACUTE ACCENT ´
- U+02BB MODIFIER LETTER TURNED COMMA ‘
- U+02BC MODIFIER LETTER APOSTROPHE ’
- U+02BD MODIFIER LETTER REVERSED COMMA ’
- U+02BE MODIFIER LETTER RIGHT HALF RING ʹ
- U+02BF MODIFIER LETTER RIGHT HALF RING ʻ
- U+02DD DOUBLE ACUTE ACCENT ˝
- U+2018 LEFT SINGLE QUOTATION MARK ‘
- U+2019 RIGHT SINGLE QUOTATION MARK ’
- U+201C LEFT DOUBLE QUOTATION MARK “
- U+201D RIGHT DOUBLE QUOTATION MARK ”
- U+2032 PRIME ′
- U+2033 DOUBLE PRIME ″

ASCII contains three of these characters (" ' `); Latin 1 contains four (" ' ` ´); Windows contains seven (" ' ` ‘ ’ “ ”); and Macintosh Roman contains eight (" ' ` ´ ‘ ’ “ ”).

(NOTE: The primes are intended to represent the measures minutes or feet, and seconds or inches, respectively. With the 8-bit Macintosh character set, I as a typographer would generally prefer APOSTROPHE ' and QUOTATION MARK " for PRIME ′ and DOUBLE PRIME ″ respectively (preferring them to ’ and ”), and I would not choose acute ´ or double acute ˝, although they are available for use in that character set, unless it were vitally important that glyphs similar to the correct prime glyphs were used – but I may have an overeducated view of encoded text, and it may be that some, or many, people have used the spacing diacritics in their coded texts.)

Conversion programs need to take these things into account. On the Macintosh, and presumably on Windows, a number of programs exist for performing transformations on text. Most of these include tables for converting Windows or EBCDIC text to Mac text or vice-versa, but they often offer other options for text conversion as well. Often these operate only on plain text files, not RTF or other formatted text. With regard to quotation marks, many programs offer mechanisms to deal with conversion of smart quotes to dumb quotes and vice versa. Eudora will convert smart quotes to dumb quotes before sending an e-mail; Quark XPress offers an option to smarten quotes when importing a text file; Nisus Writer has a built-in macro to smarten or stupefy quotation marks in an open document. These algorithms seek to pair quotes when smartening them, and sometimes present a warning to the user when unpaired marks (which appear string initially or string finally) are found. With regard to the punctuation apostrophe, the

algorithm works fine for English and French, where APOSTROPHE ' (which can be ignored in doing the pairing comparison) only occurs flanked by letters, but where the punctuation apostrophe occurs word-initially (as in Dutch and Gaelic) an incorrect transformation may occur. In conversions where those apostrophes may really represent MODIFIER LETTER APOSTROPHE ' or MODIFIER LETTER TURNED COMMA ‘ no suitable conversion is made, that is to say, no choice is made between dumb apostrophe U+0027 and U+2019, U+202C, or U+202D.

Language tagging could possibly work for Klingon in such conversions because that no vowels occur in word-initial position. Lookup with language tagging could possibly work for Dutch because the number of such strings is limited to 's, 's-, 't (and 'n for Afrikaans) and these strings are always followed by space or hyphen. For Irish Gaelic it could not, because the strings “'S é ...”, “'Sé ...” and “‘Sé ...’” could all occur (where the first two show elision and the last a quotation mark; the second and third are a minimal pair).

In Klingon, words may end in vowels or in vowels followed by glottal stops. Even lookup may not be able to resolve the ambiguity. Consider the following text, containing final MODIFIER LETTER APOSTROPHES and LEFT SINGLE QUOTATION MARKS.

ro ‘trunk of body’, *ro* ‘fist’; *cha* ‘torpedos’, *cha* ‘two, to display’; *Do* ‘velocity’, *Do* ‘be lucky, luckily’; *maw* ‘offend’, *maw* ‘be crazy’; *may* ‘be fair’, *may* ‘battle’.

Saavik reported: “When the Klingon commander said ‘wa’ yIHoh! jISaHbe’!’ the warrior killed the captain’s son.”

(Of course this gives the transliteration. In correct Klingon script the ambiguity would not occur: Saavik reported “When the Klingon commander said ‘ᳵ᳚ ᳶᳵ᳚ᳶ᳚ ᳶᳵ᳚ᳶ᳚ᳶ᳚ᳶ᳚’ the warrior killed the captain’s son.” — though even here, in a transliteration operation, where wa’ will be converted to ᳵ᳚, the ’ could prove ambiguous if it were U+2019 (cf. *may* ‘be fair’, *may* ‘battle’).

(NOTE: I have used Klingon above as an example, and this is not done as a joke. Artificial or not, features exhibited by Klingon doubtless appear in other languages. Perhaps Azerbaijani, Navajo, or Nenets share these features. I don’t know.)

John Wind’s program Add/Strip does allow the user to specify strings such as these for conversions, and presumably were it to be UCS-savvy the same mechanism could be applied.

Conversion of non-UCS data will also be complex because (as noted above) of the use of Latin 1 GRAVE for U+2018 ‘ and ACUTE for U+2019 ’. These also must be taken into account; conversion of APOSTROPHE to one or another quotation marks is not enough. I don’t know if, even for the English language, a 90% correct algorithm can be generated to handle all the transformations required.

What we need are some standards here. Word selection with regard to initial and final U+2019 (punctuation apostrophe/right single quotation mark) is needed. Standardized guidelines for plain text involving conversion of non-UCS character sets to UCS text need to be developed, and recommendations for performing such transformations, with and without language tagging and lookup, needs to be made. Possibly the algorithms could make use of the pairing principle to assist in such operations. Or possibly the

recommendations will be able to specify strategies for automatic text conversion and subsequent “things to look for” in order to inform the user of necessary post-processing search-and-replace operations. Users need to be educated about the choices they should make when the MODIFIER LETTERS are required.

References

- Betrò, Maria Carmela. 1995. *Hiéroglyphes: les mystères de l'écriture*. Paris: Flammarion. ISBN 2-08-012465-X
- Clarke, H. Wilberforce. 1878. *The Persian manual: a pocket companion intended to facilitate the essential attainments of conversing with fluency and composing with accuracy, in the most graceful of all the languages spoken in the East*. London: Wm. H. Allen & Co.
- Collier, Mark, & Bill Manley. 1998. *How to read Egyptian hieroglyphs: a step-by-step guide to teach yourself*. London: British Museum Press. ISBN 0-7141-1910-5
- Gardiner, Alan. 1966. *Egyptian grammar: being an introduction to the study of hieroglyphs*. 3rd edition. London: Oxford University Press.
- Faulmann, Carl. 1990 (1880). *Das Buch der Schrift*. Frankfurt am Main: Eichborn. ISBN 3-8218-1720-8
- Loprieno, Antonio. 1995. *Ancient Egyptian: a linguistic introduction*. Cambridge: Cambridge University Press. ISBN 0-521-44849-2
- Palmer, E. H. 1901. *The Arabic manual: comprising a condensed grammar of both the classical and modern Arabic, reading lessons and exercises, with analyses, and a vocabulary of useful words*. London: Samson Low, Marston & Co.
- UTR8. *Unicode Technical Report #8*. <http://www.unicode.org/unicode/reports/tr8.html>
- Мусаев, Кенесбай Мусаевич. 1965. *Алфавиты языков народов СССР*. Москва: Наука.