Public Review Issue #46

# Encoded representations for distinct positional uses of Hebrew Meteg

Peter Constable, Microsoft Corporation

2004-09-13

In some uses of the Hebrew script, particularly for Biblical text, a variety of combining marks are used. One of these marks is *meteg*, encoded as U+05BD, HEBREW POINT METEG.

Meteg frequently occurs together with other combining marks. When meteg co-occurs with another mark that occupies the same general space below the base character, different relative arrangements of meteg and these other marks are possible. In some uses it is considered necessary to specify these relative arrangements of meteg and other marks in the encoded representation. A proposal[1] has been submitted to UTC for how these different positionings of meteg should be specified in encoded representations. This proposal makes use of the control characters COMBINING GRAPHEME JOINER (CGJ), ZERO WIDTH JOINER (ZWJ) and ZERO WIDTH NON-JOINER (ZWNJ).

This public-review issue is soliciting feedback on this proposal and, in particular, on the proposed use of ZWJ and ZWNJ for distinguishing between the different positional uses of the meteg.

The details in this case are somewhat complex. Familiarity with combining marks, canonical combining classes, canonical ordering and canonical equivalence is assumed. Some background information on those topics is provided in an appendix.

## 1. Background: meteg in combination with below-base vowel marks

Biblical Hebrew text includes a number of marks used to annotate the text, which were introduced by Masoretic scholars over a thousand years ago. These marks include vowel points and a number of accentuation marks that indicate structural units of the text, serving to guide the reader or chanter. One of the latter marks is the *meteg*.[2]

The meteg very often co-occurs with a vowel point:

[1] L2/04-194, “On the Hebrew mark *METEG*”. This document is publicly available at http://www.qaya.org/academic/hebrew/Meteg.html.

[2] “Meteg” is used in this document to refer to a particular graphic mark. The single mark is used for distinct accentual functions for which there are separate names. These details are not crucial to the issues under consideration here, however, and so will be ignored. For further information, consult references such as Tov (1992) or Yeivin (1980).

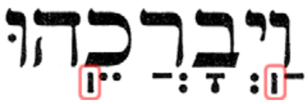

Figure 1. Meteg co-occurring with vowel points patah and tsere (BHS, 2 Sam. 13.25.15).

When the meteg co-occurs with another below-base combining mark, there is, in principle, the possibility that the two can occur in alternate positions relative to one another, as illustrated in Figure 2 and Figure 3:

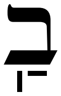

Figure 2. Meteg with patah: meteg positioned to the left of patah.

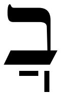

Figure 3. Meteg with patah: meteg positioned to the right of patah.

Such variations of ordering have been found to occur in Masoretic manuscripts, though there is not known to be any semantic distinction intended by Masoretic scribes.

A further aspect of this meteg-positioning issue arises in relation to three vowel marks that each consist of two side-by-side components, known as *hataf* vowels. These are shown in Figure 4:

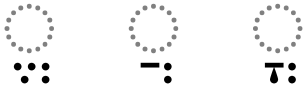

Figure 4. Hataf vowels: (from left to right) *hataf segol*, *hataf patah*, and *hataf qamats*

Because the hataf vowels consist of two components, there is a *third* logical positioning possibility: meteg may be to the left of the hataf vowel, to the right, or *between* the two components that make up the hataf vowel. Indeed, all three such variations are attested in use:

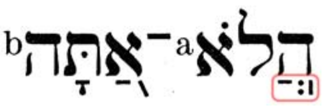

Figure 5. Meteg and hataf patah: meteg in right position (BHS, Ps. 85.7.1)

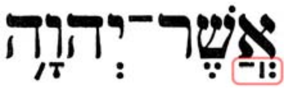

Figure 6. Meteg and hataf patah: meteg in medial position (BHS, Deut. 27.3.11)

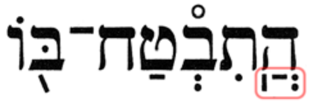

Figure 7. Meteg and hataf patah: meteg in left position (BHS, Job 39.11.1)

As has been mentioned, these variations in meteg positioning occur in Masoretic manuscripts, but no particular semantic distinction is known to have been intended by Masoretic scribes. In modern editions of the Hebrew Bible, the text is sometimes set with such variations of ordering for particular reasons, such as to preserve accurately the ordering used in certain manuscripts, or as an editorial device to reflect text-critical decisions.[3] Electronic versions of the Hebrew Bible that are most widely used by Biblical scholars are based on the *Biblia Hebraica Stuttgartensia* (BHS), an edition that has these variations; these electronic versions (eBHS) similarly preserve such ordering distinctions.

The result is that there is a textual distinction that has no semantic significance for the reading of the text and is of no interest to some users, yet for which explicit control is required by others and therefore needs to be supported in plain-text encoded representation.

## 2. The encoding problem

In general, alternate relative positioning of combining marks can be explicitly controlled in Unicode-encoded text by means of the ordering of characters in the encoded representation. There are two issues in the case of meteg, however:

- All of the vowels as well as meteg belong to separate canonical combining classes, and therefore distinct orderings of a given combination of these characters will be canonically equivalent to one another. Thus, ordering alone cannot be used as a reliable basis for representing any of the positioning distinctions for meteg. (See the appendix for additional details.)
- Given a pair of items, there are two possible orders in which they can be sequenced. When meteg is combined with the hataf vowels, however, there are three distinct positionings to be distinguished.

A partial solution to the first issue already exists, which covers the distinctions needed for meteg with non-hataf vowels. The character U+034F COMBINING GRAPHEME JOINER (CGJ) is an invisible combining mark intended for use in distinguishing otherwise-identical text elements in non-rendering text processes, particularly those that involve string comparison. It has a canonical combining class of zero, and so can be used to

[3] For instance, in the *Biblia Hebraica Stuttgartensia* (BHS), the editors chose to follow closely the positioning of the meteg on the right versus the left of a vowel used in the Leningrad Codex. In contrast, the previous editorial practice for the *Biblia Hebraica* under the direction of Rudolph Kittel (BHK) was to use the position of the meteg to indicate whether the meteg was attested in the Leningrad Codex or was inferred by the editors from other evidence.

provide a distinction between otherwise canonically-equivalent combining character sequences:

| Character sequences | Canonically-ordered representation |
|---|---|
| < ב, ◌ְ, ◌ֽ > | < ב, ◌ְ, ◌ֽ > |
| < ב, ◌ֽ, ◌ְ > | |
| < ב, ◌ֽ, CGJ, ◌ְ > | < ב, ◌ֽ, CGJ, ◌ְ > |

Table 1. Use of CGJ to distinguish right-positioned meteg

Note that, without the CGJ character, the encoded sequence < … METEG, vowel > is canonically equivalent to the sequence < … vowel, METEG > due to the canonical combining classes of the marks. By inserting the CGJ, it becomes possible to encode a canonically-ordered sequence with the METEG before the vowel. This provides a reliable and distinct representation for the right-positioned meteg in the case of non-hataf vowels:[4]

| Text element | Character sequences |
|---|---|
| בְֽ | < ב, ◌ְ, ◌ֽ > |
| בֽ͏ְ | < ב, ◌ֽ, CGJ, ◌ְ > |

Table 2. Encoded representation of left versus right meteg with non-hataf vowels

It is worth noting here that the left-positioned meteg appears to be far more frequently used than the right-positioned meteg. Thus, it is the marked (less frequent) case that has extra character, CGJ, which is a pleasing coincidence. For users that do not care about the positioning distinction, then, they do not need to use the CGJ, and the same default rendering can be provided for these uses as for those that *do* care about the positioning distinction.

The CGJ can be used with hataf vowels as well as non-hataf vowels in the representation of right-positioned meteg. This covers all of the cases for non-hataf vowel. The second issue mentioned above still remains, however: a third distinction, medial-positioned

[4] It was mentioned earlier that CGJ is intended for providing distinct results in *non-rendering* processes. The Unicode Standard states clearly that CGJ is not intended for control of rendering. One might wonder, then, why CGJ is permitted to control rendering—right versus left meteg—in this case. The answer is that it is *not* being used to control the rendering process: it is only being used to control canonical ordering and the string comparison processes used in determining canonical equivalence. It is, rather, the encoded order of METEG and the vowel marks that determines the rendering.

meteg, is needed in the case of hataf vowels. It is for this reason that a new proposal is required.

## 3. Proposal of L2/04-194: use of ZERO WIDTH JOINER and NON-JOINER

For meteg and hataf vowel combinations, with CGJ used to distinguish the right meteg from left meteg, it remains to find a representation that distinguishes *medial* meteg from left meteg.

The proposal in L2/04-194 is to consider the combination of a hataf vowel and medial-positioned meteg to be a ligature of the hataf vowel and a nominally-positioned (left) meteg. Given this conceptualization, the visual distinction can be controlled using already-defined functionality of the ZERO WIDTH JOINER (ZWJ) and ZERO WIDTH NON-JOINER (ZWNJ) for ligation control. Thus, a sequence

< …, *hataf vowel*, ZWJ, METEG >

would be used by an author to indicate that the medial-positioned meteg rendering is preferred (and would be the results presented by fonts and rendering systems that support such functionality, but could be ignored by fonts or rendering systems that do not). Likewise, a sequence

< …, *hataf vowel*, ZWNJ, METEG >

would indicate a preference for the left-positioned meteg.

The proposal in L2/04-194 recognizes that the ligation-control functionality of ZWJ and ZWNJ is one that can be supported optionally (hence the wording "preference"), and so allows that some fonts or rendering systems may ignore the joiners and display all hataf vowel-meteg combinations in a single manner. The proposal allows implementations to differ with regard to what positioning is used if the joiners are not supported: some implementations may always display meteg to the left of hataf vowels, while other implementations might always display meteg in the medial position with hataf vowels.[5] Further discussion of these issues will be continued in the next section.

L2/04-194 provides further analysis regarding the interaction between encoded representations involving meteg and other combining marks such as dagesh. These are considered because there are issues related to canonical ordering and equivalence to be considered. It is shown that the proposed solution for positional variants of meteg can work in sequences that involve these other combining characters. For further details, the reader is directed to that document.[6]

[5] Of course, the sequence < …, METEG, CGJ, *hataf vowel* > would always result in the right-positioned meteg.

[6] As mentioned above, this document is publicly available at http://www.qaya.org/academic/hebrew/Meteg.html.

# 4. Additional analysis of proposal

## 4.1 Use of joiners and default position of meteg with hataf vowels

As mentioned, the proposal document allows for the optional implementation of support for the ligation-control function of ZWJ and ZWNJ, and assumes that implementations that do not support this can choose whether to display a hataf vowel-METEG sequence with meteg in the left or medial positions.

What the proposal does not discuss, however, is what default behaviour is assumed in implementations that *do* support the ligation-control function of the joiners. That is, in a sequence without either ZWJ or ZWNJ, will the meteg appear in left or medial position? In the absence of any specification, different implementations will do different things. Thus, there may be four different kinds of implementation in relation to how sequences of a hataf vowel and meteg with or without a joiner character will be displayed:

| | Kinds of implementation | | | |
|---|---|---|---|---|
| **Character sequence** | **Type I** | **Type II** | **Type III** | **Type IV** |
| | (joiners not supported) | | (joiners supported) | |
| 1. < ..., *hataf vowel*, METEG > | left meteg | medial meteg | left meteg | medial meteg |
| 2. < ..., *hataf vowel*, ZWJ, METEG > | left meteg | medial meteg | medial meteg | medial meteg |
| 3. < ..., *hataf vowel*, ZWNJ, METEG > | left meteg | medial meteg | left meteg | left meteg |

Table 3. Four kinds of rendering implementation

The first two do not support the ligation-control function of the joiners; one always displays meteg on the left, which the other always displays medial position. The second two support the ligation-control function of the joiners, and so always display meteg in medial position if ZWJ is used, and left position if ZWNJ is used. They differ, however, in what the default rendering is when neither joiner is used: one treats the non-ligated rendering as the default, while the other treats the ligated display as the default. This situation is similar to what we might encounter in relation to *fi* and *fl* ligatures today:

| | Kinds of implementation | | | |
|---|---|---|---|---|
| **Character sequence** | **Type I** | **Type II** | **Type III** | **Type IV** |
| | (joiners not supported) | | (joiners supported) | |
| < f, i > | f + i | fi ligature | f + i | fi ligature |
| < .f, ZWJ, i > | f + i | fi ligature | fi ligature | fi ligature |
| < f, ZWNJ, i > | f + i | fi ligature | f + i | f + i |

Table 4. Four rendering implementations for fi and fl ligatures

In principle, it would be possible to specify a required default—that the rendering without either joiner must be with meteg on the left, or must be medial meteg. It is not obvious that there is a clear preference across all users for one or the other, however.

This has implications for typographers that want to distinguish among various positions of meteg. If a typographer knows that the content will only be viewed with an implementation of Type III, say, then they need only add a joiner to indicate the medial-positioned meteg (ZWJ, in that case); that is, they may use sequences 1 or 2 in their data. In another situation, in which the typographer knows that they content will only be displayed with an implementation of Type IV, then they need only add a joiner to indicate the left-positioned meteg, and may use sequences 1 and 3. In the general case, though, in which the user cannot predict whether text will be displayed using an implementation of Type III or Type IV, the only way to ensure that the meteg will appear in the desired position will be to use joiners in all cases with hataf vowels, always explicitly indicating both left and medial positions (sequences 2 and 3). That is, unless a required default was specified, ruling out one of the two types of implementation.

## 4.2 Disparate mechanisms (joiners versus CGJ) for control of meteg positioning

Considering the complete solution for control of the positioning of meteg, one possible concern is that two distinct control mechanisms are used for what can be considered a single problem: use of CGJ on the one hand, and use of ZWJ and ZWNJ on the other.

Related to this is a potential concern at the use of ZWJ and ZWNJ for this purpose. For instance, a proposal to control positioning of the holam dot over vav using joiners was recently rejected by UTC, one of the reasons being that it was considered undesirable to specify the use of ZWJ or ZWNJ for making semantic distinctions (the two positions of holam being associated with distinct readings of the text). In the case of meteg, however, UTC is open to the possible use of ZWJ and ZWNJ since the alternate positions of meteg are purely presentational: there is no know semantic distinction associated with the difference between left- and medial-positioned meteg.

Of course, the distinction between right- and left-positioned meteg, indicated using CGJ, is no more or less semantic than that between medial- and left-positioned meteg. All are simply matters of presentation, yet the default behaviours for various text processes will be to ignore the joiners but not CGJ, treating left meteg different from right meteg, but not medial meteg. These are default behaviours, however, and tailored process can be written that treat all of these meteg sequences in a similar manner.

Thus, in terms of valid application of Unicode design principles or in terms of text processing behaviours, there is not any serious obstacle to adopting the proposed solution. Some may still be concerned that it is inelegant to use multiple mechanisms for what is perceived to be a single type of problem. A possible response to that is that it is not essential to have a maximally elegant solution to a minor problem occurring in relatively very limited usage.

## 4.3 Overall user impact

In adding any new encoding specification, it is desirable that there be the least negative impact on users. In particular, among Hebrew users, the vast majority is not particularly concerned about the distinctions in meteg positioning, and they should not be burdened with additional costs as a result of changes to support control of meteg positioning.

Usage of Hebrew script can be broadly categorized in three ways:

- Usage that does not involve the Masoretic accentuation system, including meteg. This covers the vast majority of usage.
- Usage that involves meteg, but in which distinctions in its position is considered unimportant. This covers a relatively very limited though still significant number of users.
- Usage that involves meteg and in which distinctions in its position are of interest. This is the most limited category in terms of number of users.

For users in the first category, since meteg is not even used, users are not at all affected by these changes. In the third category, of course users are affected, but not in negative ways. Users in the second category are not directly affected by these changes: they can continue to use the encoded representations they have always used. There is potential for concern when users in the second and third categories come into contact, however.

Consider two users, A and B: user B cares about the distinct positions of meteg, while user A does not. If user A prepares content that is received by user B, then B must deal with the fact that the content will not make distinctions that B cares about. Going the other direction, if B prepares content that is received by A, A may encounter unexpected results when processing the text; for instance, a spelling checker might not accept a sequence that includes CGJ. Fortunately, the cases in which CGJ, ZWJ or ZWNJ might be used are rare,[7] so these costs to users are limited. These must simply be considered costs of introducing a distinction between the different forms of meteg; they would exist regardless of what solution was provided.

[7] In the eBHS text, the case of right meteg with a non-hataf vowel or meteg with a hataf vowel account for less than 3% of all uses of meteg.

## Appendix A. Combining marks and canonical ordering

Unicode allows for text elements to be composed dynamically by addition of combining-mark characters in combining character sequences. Any given combining mark occupies some nominal position relative to its base — centered below, for instance — and when multiple combining marks are combined in a single combining character sequence, issues arise regarding how the marks interact typographically (that is, positionally) one with another. In particular, if two marks occupy distinct positions relative to the base, then they do not interact, and so distinct encoded sequences in which the characters are ordered differently are visually indistinct:

| Text element | Character sequences |
|---|---|
| ɛ̤̆ | < U+025B LATIN LETTER SMALL OPEN E,<br>U+0324 COMBINING DIAERESIS BELOW,<br>U+0306 COMBINING BREVE > |
| ɛ̤̆ | < U+025B LATIN LETTER SMALL OPEN E,<br>U+0306 COMBINING BREVE,<br>U+0324 COMBINING DIAERESIS BELOW > |

Table 5. Combining character sequences that are *visually indistinct*

In contrast, if the two combining marks occupy the *same* position relative to the base, as in the case of two center-below marks, there is an interaction between these marks. A principle applicable to many, though not all, cases is used whereby marks are stacked vertically outward from the base in the order in which they occur:

| Text element | Character sequences |
|---|---|
| ɛ̤̥ | < U+025B LATIN LETTER SMALL OPEN E,<br>U+0324 COMBINING DIAERESIS BELOW,<br>U+0325 COMBINING RING BELOW > |
| ɛ̥̤ | < U+025B LATIN LETTER SMALL OPEN E,<br>U+0325 COMBINING RING BELOW,<br>U+0324 COMBINING DIAERESIS BELOW > |

Table 6. Combining character sequences that are *visually distinct*

There is a correlation to be noted: where two marks occupy the same position relative to the base, two alternately-ordered combining character sequences containing those two marks are visually distinct and so can be used with distinct linguistic meanings; but if two marks occupy different positions relative to the base, then the two alternately-ordered character sequences are visually indistinct. This correlation is captured in part by the assignment of combining marks to *canonical combining classes*: in the normal case, if two marks occupy a similar position relative to the base, they are assigned to the same canonical combing class. These classes are used in defining canonical equivalence relationships between sequences of characters. Thus, for the examples in Table 5, the fact that the marks belong to different classes results (following the relevant definitions) in the two character sequences being considered *canonically equivalent*, whereas for the examples in Table 6, the fact that the marks belong to the same class results in the two sequences *not* being considered canonically equivalent.

As mentioned above, the principle of stacking marks vertically outward from the base applies in many but not all cases. In particular, in Hebrew script, combinations of marks sharing the same nominal position relative to the base are generally positioned side-by-side.

The use of canonical combining classes in defining canonical equivalence still applies, however: alternately-ordered sequences are canonically equivalent if the marks are in distinct classes, but are not canonically equivalent if they are in the same class. So, for instance, in Table 7, the two Hebrew marks are in distinct classes, and accordingly the two alternately-ordered sequences are visually indistinct:

| Text element | Character sequences |
|---|---|
| א֤֙ | < U+05D0 HEBREW LETTER ALEF,<br>U+05A4 HEBREW ACCENT MAHAPAKH,<br>U+0599 HEBREW ACCENT PASHTA > |
| א֤֙ | < U+05D0 HEBREW LETTER ALEF,<br>U+0599 HEBREW ACCENT PASHTA,<br>U+05A4 HEBREW ACCENT MAHAPAKH > |

Table 7. Hebrew combining character sequences that are *visually indistinct*

These examples are no different from the analogous Latin examples in Table 5. In the examples in Table 8, however, the two marks belong to the same class. When they co-occur, they position side-by-side, but the two sequential orderings correspond to two distinct appearances:

| Text element | Character sequences |
|---|---|
| ז֜֠ | < U+05D6 HEBREW LETTER ZAYIN,<br>U+059C HEBREW ACCENT GERESH,<br>U+05A0 HEBREW ACCENT TELISH GEDOLA > |
| ז֠֜ | < U+05D6 HEBREW LETTER ZAYIN,<br>U+05A0 HEBREW ACCENT TELISH GEDOLA,<br>U+059C HEBREW ACCENT GERESH > |

Table 8. Hebrew combining character sequences that are *visually distinct*

In this way, the role of canonical combining classes in determining canonical equivalence relationships is the same for Hebrew as for any other script. There is one special consideration in the case of Hebrew, however: whereas, in the normal situation described above, marks that occupy a similar position relative to the base are assigned to the same class, several of the below-base marks in Hebrew are assigned to distinct classes.[8] Some of these are described in Table 9:

| Character | Canonical combining class[9] |
|---|---|
| ◌ְ U+05B0 HEBREW POINT SHEVA | 10 |
| ◌ֱ U+05B1 HEBREW POINT HATAF SEGOL | 11 |
| ◌ֲ U+05B2 HEBREW POINT HATAF PATAH | 12 |
| ◌ִ U+05B4 HEBREW POINT HIRIQ | 14 |
| ◌ָ U+05B8 HEBREW POINT QAMATS | 18 |
| ◌ֽ U+05BD HEBREW POINT METEG | 22 |

Table 9. Hebrew below-base marks in distinct canonical combining classes

One result of this is that alternately-ordered sequences involving combinations of these marks will be canonically equivalent. For instance, the following sequences are canonically equivalent:

< ב U+05D1 HEBREW LETTER BET, ◌ְ U+05B0 HEBREW POINT SHEVA,
◌ֽ U+05BD HEBREW POINT METEG >

< ב U+05D1 HEBREW LETTER BET, ◌ֽ U+05BD HEBREW POINT METEG,
◌ְ U+05B0 HEBREW POINT SHEVA >

The significance of these sequences being considered canonically equivalent is that processes cannot assume any distinction between them; in practice, most processes should treat them as identical.

As a result, even though these marks occupy similar positions relative to the base, different visual configurations of the marks cannot be reliably represented by means of distinct orderings of the characters in encoded representation. This is a critical point in the case of meteg, since different visual configurations of meteg with other below-base marks need to be distinguishable in encoded representations. Since ordering of encoded character sequences alone is not sufficient, other encoding mechanisms are required.

As further background to the meteg issue, it is important to understand the mechanism by which canonical combining classes are used to determine canonical equivalence

[8] This is the case for vowel marks and meteg. The reasons for these combining class assignments have no particular relevance for this discussion. A point that is relevant, though, is that these assignments cannot be changed.

[9] Each canonical combining class is assigned a unique integer value. This integer value is a mechanism used in determining a canonically-ordered representation for combining character sequences, which is used as a basis for comparison in the determination of canonical-equivalence relationships between sequences, as explained below.

relationships between sequences.[10] Each canonical combining class is assigned an integer value, as shown for the examples in Table 9. These integer values are used to define a canonical ordering of combining marks within a combining character sequence: for any combining character sequence, there is an equivalent *canonically-ordered* representation in which sequences of combining marks belonging to non-zero classes occur in order according to the integer values of their classes. To illustrate, Table 10 shows some example sequences and their corresponding canonically-ordered representation, based on the classes shown above in Table 9:

| **Example character sequence** | **Corresponding canonically-ordered representation** |
|---|---|
| < ב, ◌ְ, ◌ֽ > | < ב, ◌ְ, ◌ֽ > |
| < ב, ◌ֽ, ◌ְ > | < ב, ◌ְ, ◌ֽ > |
| < ב, ◌ֽ, ◌ָ, ◌ִ > | < ב, ◌ִ, ◌ָ, ◌ֽ > |
| < ב, ◌ִ, ◌ֽ, ◌ָ > | < ב, ◌ִ, ◌ָ, ◌ֽ > |
| < ב, ◌ָ, ◌ֽ, ◌ִ > | < ב, ◌ִ, ◌ָ, ◌ֽ > |

Table 10. Example combining character sequences and their canonically-ordered representations

Note for the first two example sequences that the corresponding canonically-ordered representations are the same. This determines that these two example sequences are canonically equivalent. Similarly, the last three example sequences are canonically equivalent since their corresponding canonically-ordered representations are identical.

It is important to note the effect of combining characters belonging to the zero class: when they occur within a combining character sequence, they divide the sequence such that the integer-based ordering is applied independently to each sub-sequence between the "zeroes"; that is, no re-ordering occurs around a character with a combining class of zero.

For instance, consider a hypothetical character $x$ with a combining class of zero. The effect of such a character on canonical ordering when it occurs within a combining character sequence is illustrated in Table 11 (the integer values of the classes of each combining mark are added following the character sequences as an aid to understanding):

[10] The explanation provided here uses a slightly different description than that used in the formal definition of canonical ordering within the Unicode Standard, though what is described is consistent with what is specified in the Standard. For further details, including the formal definitions of *canonical equivalence* and *canonical ordering*, see §3.7 and §3.11 of Unicode 4.0.

| Example character sequence | Corresponding canonically-ordered representation |
|---|---|
| < ב, ◌ֽ, ◌ְ > (22, 10) | < ב, ◌ְ, ◌ֽ > (10, 22) |
| < ב, ◌ֽ, $x$, ◌ְ > (22, 0, 10) | < ב, ◌ֽ, $x$, ◌ְ > (22, 0, 10) |
| < ב, ◌ֽ, ◌ָ, ◌ִ > (22, 18, 14) | < ב, ◌ִ, ◌ָ, ◌ֽ > (14, 18, 22) |
| < ב, ◌ֽ, $x$, ◌ָ, ◌ִ > (22, 0, 18, 14) | < ב, ◌ֽ, $x$, ◌ִ, ◌ָ > (22, 0, 14, 18) |
| < ב, ◌ֽ, ◌ָ, $x$, ◌ִ > (22, 18, 0, 14) | < ב, ◌ָ, ◌ֽ, $x$, ◌ִ > (18, 22, 0, 14) |

Table 11. Effect of class-zero marks on canonical ordering

So, for instance, in the second set of examples, the last combining mark (hiriq) belongs to the class with the smallest integer value. Thus, while it occurs first in the canonically-ordered representation when the character $x$ is not present (see the first of the three examples), it is blocked from re-ordering in the last example by the presence of $x$. Since the canonically-ordered representations for the last two examples are distinct, those two sequences are not canonically equivalent.

## 5. References

Elliger, K.; W. Rudolph; and A. Schenker; eds. 1997. *Biblia Hebraica Stuttgartensia, 5th revised edition.* Stuttgart: Deutsche Bibelgesellschaft. (“BHS”)

Kittel, Rudolph, ed.. 1952. *Biblia Hebraica 3, 7th edition.* Stuttgart: Privileg. (“BHK”)

Tov, Emmanuel. 1992. *Textual criticism of the Hebrew Bible, 2nd revised edition*. Minneapolis: Fortress Press.

Yeivin, Israel. 1980. *Introduction to the Tiberian Masorah*. (Masoretic studies, 5.) Translated and edited by E.J. Revell. Scholars Press.