L2/01-415
2 November 2001

Interaction of Ligation Control via High-Level Protocols and ZWJ

John H. Jenkins
Apple Computer, Inc.
jenkins@apple.com

The Unicode Standard, version 3.1, includes the following change to §13.2:

> The intended semantic [of `U+200D ZERO WIDTH JOINER`] is to produce a more connected rendering of adjacent characters than would otherwise be the case, if possible. In particular:
>
> 1. If the two characters could form a ligature, but do not normally, ZWJ requests that the ligature be used.
>
> 2. Otherwise, if either of the characters could cursively connect, but do not normally, ZWJ requests that each of the characters take a cursive-connection form where possible.
>
> * In a sequence like <X, ZWJ, Y>, where a cursive form exists for X, but not for Y, the presence of ZWJ requests a cursive form for X.
>
> 3. Otherwise, where neither a ligature nor cursive connection are available, the ZWJ has no effect.

It further states:

> For modern font technologies, such as OpenType or AAT, font vendors should add ZWJ to their ligature mapping tables as appropriate. Thus where a font had a mapping from "f" + "i" to fi, the font designer should add the additional mapping from "f" + ZWJ + "i" to fi. On the other hand, ZWNJ will normally have the desired effect naturally for most fonts without any change, since it simply obstructs the normal ligature/cursive connection behavior. As with all other alternate format characters, fonts should use an invisible zero-width glyph for representation of both ZWJ and ZWNJ.

The original language of the UTC resolution dictating this change also included a notice that this mechanism for ligature control may not be best suited for Latin typography. This notice was omitted for some reason in the language used in Unicode 3.1. The editorial committee is working on rewording to reflect this defect, and one hopes that it

will be included in Unicode 3.2.

There is, however, an additional concern.

Even for default ligatures like “fi” and “fl,” systems which implement ligature control for Latin typography such as Adobe’s **InDesign** and Apple’s MLTE and Cocoa programming interfaces allow even default ligatures to be turned on and off. One can either write “difficult fleshy fish” or “difficult fleshy fish” at will. This is true even if the ZWJ is present to dictate ligature control.

The intent of allowing ZWJ as a ligator control is to handle those situations where a ligature is linguistically necessary and its presence or absence can signal a change in meaning. Given this, it would appear that in such instances, <X ZWJ Y> should form the <XY> ligature if possible *even if the user has otherwise turned off ligatures*. This is particulraly true since the current version of the text expressly notes that inserting a ZWNJ will break ligatures automatically even if ligation is turned on.

(We note that in any event, we’ve got the awkward situation developing where the originator of a text may turn ligatures on via one mechanism and the receiver attempt to turn them off via the other one, a situation fraught with potential for frustrated end users.)

Apple requests that the UTC clarify the interaction of ZWJ and higher-level ligature control protocols. Language should be added to §13.2 of the language to emphasize that ZWJ for use as a ligature control should only be used in cases where the choice between having a ligature or not having it is a *linguistic* and not a *stylistic* one, and the section should clearly indicate whether or not ZWJ as a ligature control overrides other ligature control protocols.