Public Review Issues

181	Changing General Category of Twelve Characters	2011.05.02
Status:	Open	

Description of Issue:

The UTC has decided to change the general category of twelve characters. The characters in question are these:

```
U+00AA FEMININE ORDINAL INDICATOR
U+00BA MASCULINE ORDINAL INDICATOR
U+1D62 LATIN SUBSCRIPT SMALL LETTER I
U+1D63 LATIN SUBSCRIPT SMALL LETTER R
U+1D64 LATIN SUBSCRIPT SMALL LETTER U
U+1D65 LATIN SUBSCRIPT SMALL LETTER V
U+1D66 GREEK SUBSCRIPT SMALL LETTER BETA
U+1D67 GREEK SUBSCRIPT SMALL LETTER GAMMA
U+1D68 GREEK SUBSCRIPT SMALL LETTER RHO
U+1D69 GREEK SUBSCRIPT SMALL LETTER PHI
U+1D6A GREEK SUBSCRIPT SMALL LETTER CHI
U+2C7C LATIN SUBSCRIPT SMALL LETTER J
```

The UTC intends to change the general category of these characters from its current value of "Ll" to the value "Lm". The rationale is that superscript or subscript letters with decompositions to a single character should consistently have gc=Lm. Changing the general category for these twelve characters aligns them with the 122 other superscript or subscript letters whose General_Category is already "Lm".

This change for the General Category property implies some changes for dependent casing properties. In particular, in order to keep the derived Lowercase property values unchanged, each of the twelve characters will have the contributory property Other_Lowercase set to Yes. The property Case_Ignorable, which is a narrow-use property only relevant to some special casing boundary determination (see D136 and Table 3-15 in Chapter 3 of Unicode 6.0 for details), would change from No to Yes for these twelve characters. The changes are summarized in the following table:

Property	Old Value	New Value
General_Category	Ll	Lm
Other_Lowercase	No	Yes
Lowercase	Yes	Yes
Case_Ignorable	No	Yes

The behavior of software may change for these twelve characters if it is dependent on a distinction between gc=Ll versus gc=Lm, or on the value of the Case_Ignorable property.

Feedback is being requested on the positive and negative effects, if any, these changes would have on existing implementations. A change in behavior may be considered positive, for

example, if it results in a more uniform treatment of compatibility super/subscript characters and modifier letters. It may be considered negative if the change in properties produces an unexpected result or forces an unwanted change to software to compensate for the change.