UTC/2000-004

Proposal for UTC: Control Codes

Mark Davis, Ken Whistler
January 12, 2000

Mark has suggested a number of changes to normative General Category property values for some control codes -- namely the ones that have directional property values for the bidi algorithm other than the default BN.

I am very, very leery of this suggestion, for several reasons.

First, this is fiddling with a normative General Category property for a well-defined set of entities: Cc for the set of ISO control code values. As it currently stands, Cc applies to U+0000..U+001F and U+007F..U+009F, and to *only* those ranges. Those ranges should look familiar, of course, since they exactly match the control code values dealt with in the ISO 2022 framework and in ISO 6429. Bleeding off the values of characters given the Cc assignment in favor of some other normative General Category would leave the rump set of control codes assigned Cc an unprincipled collection of things. Effectively this proposal would change the *meaning* of the Cc category. It would definitely impact implementations, since I know of at least one library and a whole set of related utilities that depend on these values as they are currently defined. :-)

Second, I cannot make a whole lot of sense of the *particular* reassignments that Mark is proposing. To see why I think this is a problem, I list here the larger property context that this set of characters exists in currently.

Char	GenCat	Bidi	Space	WSpace	Delimiter	Proposed
PS	Zp	B	-	x	x	
LS	Zl	WS	-	x	x	
SPACE	Zs	WS	x	x	x	
NBSP	Zs	CS	x	x	x	
ZWSP	Zs	BN	x	x	x	
TAB,VT	Cc	S	-	x	x	==> Zs
US	Cc	S	-	-	x	==> Zs
LF,CR,NL Cc	B	-	x	x	==> Zl	
FS,GS,RS Cc	B	-	-	x	==> Zl	
FF	Cc	WS	-	x	x	==> Zl

GenCat and Bidi are the normative General Category values and Bidirectional property values defined in UnicodeData.txt.

Space, W(hite)Space, and Delimiter are the character properties that I have made available in the supplementary (and informative) file, PropList.txt. In that file, "Space" is closely and intentionally synched to the General Category "Zs", so it is not an independently derived property. "WhiteSpace" is intended to do a decent job of indicating what set of characters would be considered whitespace by a typical text parser, say of program text. It includes all "Space", Zp and Zl, and all format control codes that typically appear in such text, i.e. those that get bidi properties of "B" or "S", except for the mostly obsolete tape record delimitation control codes that don't appear in text. "Delimiter" is a much broader concept that includes both invisible delimiters such as spaces and text delimiting format control codes, but also visible delimiting punctuation and the like: basically characters that tend to mark the edges of chunks of something.

Now let's look at what Mark is proposing to change the General Category of the listed set of Cc's to.

A. "Tabs" go from Cc to Zs. I find this problematical, since it muddies the meaning of Zs. Currently, Zs is pretty carefully aligned with the list of characters that people would claim are formally "spaces". This change would mean that Zs would henceforth mean "spaces or tabs" -- a change in the meaning of Zs that could potentially impact existing implementations. Furthermore, this change does *not* result in any particularly better matching between the General Category and the Bidi property, contrary to Mark's claim that "this will bring the general category in line with the BIDI property". Bidi B could be either Zp or Zs, and General Category Zs could be either Bidi B, WS, CS, or BN. I see no gain there over the current situation.

B. "Line breaks" and FF go from Cc to Zl. This makes a little more sense to me, at least for the line breaks, since it would align the line breaks with LS. But it makes less sense for FF, which gets treated as WS for Bidi. And the alignment between the General Category and the Bidi category is not particularly good here, either. Zl can be Bidi B or WS. Bidi WS can be General Category Zl or Zs. No big gain there, either.

So, given the marginal gains I can see here (if any), I am really inclined not to rock the boat on this (and cause problems for existing implementations) on the dubious hope that changing the General Category will make people make fewer mistakes in their programming for special handling of control format characters.

Frankly, I see this proposal as special pleading to change the relative importance of interpretation for two (or more) categories of character attributes that don't divide neatly into a partition: ISO control code status versus Unicode-specified control format function. This is just one more limitation of the General Category field, and shouldn't be addressed by tinkering with normatively defined properties in this way.

--Ken

>
> Due to changes surrounding the merger with 10646, the UTC refrained from
> assigning names and properties to control codes. In theory, implementions
> are free to assign whatever meaning to whatever control codes that they
> want.
>

```
> In practice, this is simply counter-productive. We already break this
> practice with regard to BIDI properties. For the general category, we have
> seen a number of implementers just pick up the properties, then be
> surprised when the common control codes have odd behavior, especially
> regarding isWhiteSpace.
>
> I propose that we change the general category of certain control codes
> (those currently having BIDI properties) to have a specific general
> property, namely setting LF, CR, FF, NL to Zl (Separator, Line) and the
> others to Zs (Separator, Space). The characters in question are the
> following (with their old general categories).
>
> 0009;<control>;Cc;0;S;;;;;N;HORIZONTAL TABULATION;;;;
> 000A;<control>;Cc;0;B;;;;;N;LINE FEED;;;;
> 000B;<control>;Cc;0;S;;;;;N;VERTICAL TABULATION;;;;
> 000C;<control>;Cc;0;WS;;;;;N;FORM FEED;;;;
> 000D;<control>;Cc;0;B;;;;;N;CARRIAGE RETURN;;;;
>
> 001C;<control>;Cc;0;B;;;;;N;FILE SEPARATOR;;;;
> 001D;<control>;Cc;0;B;;;;;N;GROUP SEPARATOR;;;;
> 001E;<control>;Cc;0;B;;;;;N;RECORD SEPARATOR;;;;
> 001F;<control>;Cc;0;S;;;;;N;UNIT SEPARATOR;;;;
>
> 0085;<control>;Cc;0;B;;;;;N;NEXT LINE;;;;
>
> This will bring the general category in line with the BIDI property, and
> reduce the opportunity for error with implementers.
>
> Mark
```