RTF Miscellany

Paul DuBois
dubois@primate.wisc.edu

Wisconsin Regional Primate Research Center
Revision date: 4 April 1991

Introduction

This document contains a few scribblings about things which don't seem to be covered in the RTF specification. Any or all conjectures here may be false; if so, I'd like to know about it. Counterexamples or references to corrections would be appreciated, as my conclusions are based on observation.

Nomenclature:

WfM Word for Macintosh
WfW Word for Windows

Page Orientation

WfM and WfW write the paper width and height correctly, but don't write ''\landscape'' into landscape documents. I consider this a bug. Another bug is that if you read such an RTF document back into WfM (and maybe WfW), it sets the page orientation back to portrait.

Tab Handling

When a new group begins (with ''{''), it inherits the tab stops of the group within which it occurs. However, if any tabs are explicitly set within the group, they override the inherited set.

The same is true with regard to tabs stops that a group may get as a result of a style setting. If tabs are set within the style, they override inherited tabs. Again, however, if tabs are explicitly set within the group, they override not only any inherited tabs, but any tabs that may have been set within the style.

For translator purposes, the upshot is that it's necessary to know when a style is being expanded (so you know to override inherited tabs), and when the expansion is done (so you know when to override style tabs).

Tabs may be associated with a leader character, and may have a justification attribute. I have seen an RTF specification (non-Microsoft) which claimed that justifications apply to the last specified tab position. My experience is that the opposite is true. (The Microsoft spec seems to be silent on this point.) The leader character does indeed apply to following tab positions, but the justification attribute applies to the next tab position specified. If no justification is given, the tab defaults to left-justified.

For translators, this means that if a justification indicator occurs, you apply it when the next tabstop position is given, otherwise the tabstop is left-justified.

Styles

The ''Normal'' style definition does not seem to ever include a style number within it, unlike all others. I assume it's style 0, since other styles that are based on the Normal style have ''\sbasedon0'' as part of their definition. The Normal style also includes the control word ''\sbasedon222'', but style 222 is never defined. Presumably this is a way of indicating ‘‘not based on any other style,’’ although why the value 222 was chosen is beyond me.

Stylesheet entries may also leave out the ‘‘\sbasedon’’ control word (WfW does this). In this case, the default should be 222. I’ve only seen this in Normal style entries.

I have not seen any documents where the ‘‘\snext’’ control word is left out, but if it were, presumably the style should be its own next style.

Restoring Defaults

Apparently, when ‘‘\pard’’ is encountered, the way to restore paragraph defaults is to restore not only all the static initial paragraph formatting values, but also to apply the Normal style.

The control word ‘‘\plain’’ is much like ‘‘\pard’’ but for characters. Why it’s not ‘‘\chard’’ I don’t know, but the effect is not only to restore character style to plain, but also default font size, expansion value, etc.

Special Characters

Different character sets may be specified in RTF documents (e.g., ‘‘\ansi’’, ‘‘\mac’’), and characters within one set may have no representation in another set. For instance, the Apple character in the Macintosh character set has no counterpart in the ANSI character set. This constitutes a failure of RTF to provide machine independence.

Fonts

The default font (specified with ‘‘\deff’’) need not be included in the font table. Don’t assume it will be there.

The font name for a given font may vary among systems. ‘‘Times Roman’’ in WfM is ‘‘Tms Rmn’’ in WfW. The font family may also differ. WfM associates ‘‘\ftech’’ with the Symbol font, whereas WfW associates it with ‘‘\fdecor’’. Another failure of machine independence.

Presumably the font name differences are related to the method (or lack of it) provided by underlying system software for referring to fonts from within programs. Translators, which may be writing output for target systems having an entirely different set of conventions, should provide their own mechanism for mapping RTF font names onto the fonts that are available.

It is instructive to observe that WfM and WfW do not do particularly well at figuring out the fonts used in an RTF document created by each other.

Tables

The Microsoft RTF specification says very little about the constraints on the order in which table formatting control words may appear. The inferences below are based on inspection of several RTF tables, but as this is an inductive process, it’s hard to say whether they are generally true, or only true of those files at which I’ve looked.

Cells are like tabs in that a cell may have a position and other attributes. The position applies to the right edge. The other attributes, if specified, occur *before* the position specifier.

In the table layout information at the beginning of the table, cell border control words are *followed* immediately by a border type control word. This is one place where a translator may want to read ahead in the input stream to get the border type. It appears that more than one border type word may follow the cell border control word, ‘‘\clbrdrb\brdrsh\brdrs’’.

It appears that (i) ‘‘\trowd’’ occurs as the first control word of a table; (ii) the row ends with ‘‘\row’’; (iii) everything between specifies the format and content of the row. It further appears that the table ‘‘state’’ is completely independent of the grouping level.

Cells begin with the *first* ‘‘\intbl’’ and after *each* ‘‘\cell’’ control word. Cells end after each ‘‘\cell’’ and ‘‘\row’’ control word. Cells can *not* be assumed to begin with every ‘‘\intbl’’ since every paragraph within a table cell begins with that control word.

Tabs specified within a cell are relative to the left edge of the cell, not the left margin of the page.

WfM seems to write an empty cell at the end of each row. For instance, if there are three cells, there will be three token sequences, each ending with ‘‘\cell’’, then another sequence (typically ‘‘\pard \intbl’’) ending with ‘‘\row’’. This probably corresponds to the ghost column you see at the end of tables when you select them or show formatting codes.