## TEdit-Line-Numbering

By: Randy Trigg (Trigg.pa)

This document last edited on May 9, 1988

**INTRODUCTION**

TEDIT-LINE-NUMBERING enables the automatic conversion of a TEdit selection into multiple lines of specified width each ending in a carriage return and prefixed by a line number.   For example, TEDIT-LINE-NUMBERING converted the following piece of transcript:

*--------------*

*C:      We have to be able to check that, within the memory I, I claim.*

*(1.0)*

*M:      Check whether Tore is a graduate student? I think we can do that (.) I mean*

*C:      Yea I know but more we have to be able to, within the memory somehow, recognize that (.) because of this constraint being in the memory, we have to check that the time matches Wednesday morning.  We have to add this constraint to the time and see if (.) the time is not overconstrained.*

*--------------*

into:

*--------------*

*021    C:      We have to be able to check that, within the memory I, I claim.*

*022    (1.0)*

*023    M:      Check whether Tore is a graduate student? I think we can do that*
*024            (.) I mean*

*025    C:      Yea I know but more we have to be able to, within the memory*
*026            somehow, recognize that (.) because of this constraint being in*
*027            the memory, we have to check that the time matches Wednesday*
*028            morning.  We have to add this constraint to the time and see if (.)*
*029            the time is not overconstrained.*

*--------------*

**INTERFACE**

(MAKE-LINE-BREAKS *TextStream WidthInInches Device StartLineNum LineNumDigits FirstLineParaLooks OtherLineParaLooks InsertExtraTabFlg InsertExtraCRFlg*)

TextStream should be a TEXTSTREAM of an open TEdit. Line breaks and line numbers will be inserted for the text in the current selection. WidthInInches is the number of inches wide the resulting lines will be when printed to Device. Device should be one of the litatoms DISPLAY, INTERPRESS, or PRESS. StartLineNum will be the line number used for the first line formed. LineNumDigits is the number of digits used to print the line numbers. FirstLineParaLooks should be a standard TEdit paralooks proplist for the first line formed from every paragraph. OtherLineParaLooks will be used for the remaining lines of each paragraph. InsertExtraTabFlg, if non-nil, will cause an extra tab to be inserted after the line number. InsertExtraCRFlg, if non-nil, causes an extra carriage return to be inserted between paragraphs. (If this last Flg is on, then FirstLineParaLooks and OtherLineParaLooks are probably equal.)

For example, the conversion depicted above was done with the following call (where SS is bound to an open textstream):

(MAKE-LINE-BREAKS SS 5 'INTERPRESS 21 3 '(LINELEADING 2 PARALEADING 5) '(LINELEADING 2 PARALEADING 0) T)

The algorithm used is (pretty much) as follows: First, place a temporary marker at the end of the selection. For each paragraph in the selection, insert a line number LineNumDigits wide followed by a tab. Move one character at a time through the paragraph adding up STRINGWIDTHs until reaching WidthInInches (converted to appropriate units for Device). Then move back to last whitespace and insert a carriage return. Change the paralooks of the paragraph just formed according to FirstLineParaLooks. Insert the next line number followed by one or two tabs (depending on value of InsertExtraTabFlg) and continue adding up STRINGWIDTHs. When WidthInInches is reached, insert a carriage return and change the paralooks to OtherLineParaLooks. Continue in this manner till the end of the paragraph. If InsertExtraCRFlg is non-nil, then insert an extra carriage return. Continue with the rest of the paragraphs until encountering the marker at the end of the selection. Delete the marker and quit.

**NOTES**

It's a good idea to have a couple of tabs set for the selected text, though MAKE-LINE-BREAKS will use the default tab setting if you don't.

It tries to do proper measurement of embedded tabs, but this hasn't been extensively tested.

**BUGS**

IT'S INCREDIBLY SLOW! This is because we use only calls available from the TEdit programmers interface. Things could be significantly sped up by walking the piece table like TEdit does when printing, but it's alot of work to figure out how to do that and anyway I'd prefer that this tool only call advertised functions.

Doesn't handle imageobjs.