

Mechanisms in fm2endnote5

The process must remove the string portions or lines which are bolded in the following examples.

The left brace is the trigger character - this is not used in ordinary mif.

File header

From `<MIFFile 7.00> # Generated by FrameMaker 7.0p578`

To `<MIFFile 6.00> # FM7-mif modified by mif2endnote [DDD]`

Ordinary use of braces

Input

Ordinary use of braces {whatever} must not use the same pattern as the temporary citations {value, parameter1 #1} stans stans

From

```
<ParaLine
  <String `Ordinary use of braces {whatever} must not use the same pattern '`
  > # end of ParaLine
<ParaLine
  <String `as the temporary citations {value, parameter1 #1} stans stans '`
  > # end of ParaLine
```

To

```
<ParaLine
  <String `Ordinary use of braces {value, parameter1 #1} stans stans '`
  > # end of ParaLine
```

Currently the first case with no comma and # in the text will cause fm2endnote5 to delete the text between the first opening brace and the next one.

The resulting construct will be reported as non-matching temporary citation byEndnote.

Ordinary entry

Input

cinctra latrantium canum turma {Garneau, 1990 #12}.

From

```
<ParaLine
  <String `cinctra latrantium canum turma {Garneau, 1990 #12}.'>
  > # end of ParaLine
```

To

```
<ParaLine
  <String `cinctra latrantium canum turma {#12}.'>
  > # end of ParaLine
```

Split entry

Input

stans stans stans split entry at end of this paragraph
{LaBonté, 1990 #8}.

From

```
<ParaLine  
  <String `stans stans stans split entry at end of this paragraph {LaBont\x8e , '>  
> # end of ParaLine  
<ParaLine  
  <String `1990 #8}.'>  
> # end of ParaLine
```

To

```
<ParaLine  
  <String `stans stans stans split entry at end of this paragraph {#8}.'>  
> # end of ParaLine
```

Highlighted entry

Input

et iusto odio dignissim qui blandit **{Bartels, 1981 #14}** praesent lup

From

```
<ParaLine  
  <String `et iusto odio dignissim qui blandit'>  
  <Char HardSpace>  
  <Font  
    <FTag `name'>  
    <FLocked No>  
  > # end of Font  
  <String `{Bartels, 1981 #14}'>  
  <Font  
    <FTag ``'>  
    <FLocked No>  
  > # end of Font  
  <String ` praesent lup'>  
  <Char SoftHyphen>  
> # end of ParaLine
```

To

```
<String `{#14}'>
```

Highlighted elements in entry

Input

tatum zrril delenit augue dui dolore te feugait nulla facilis
{Bartels, 1981 #14}, {Daube, 1989 #6}, {LaBonté, 1989 #11}

From

```
<ParaLine
  <String `tatum zrril delenit augue dui dolore te feugait nulla facilis {'>
  <Font
    <Ftag `2-highlight'>
    <FLocked No>
  > # end of Font
  <String `Bar'>
  <Char SoftHyphen>
> # end of ParaLine
<ParaLine
  <String `tels'>
  <Font
    <Ftag ``'>
    <FLocked No>
  > # end of Font
  <String `, 1981 #14}, {'>
  <Font
    <Ftag `1-highlight'>
    <FLocked No>
  > # end of Font
  <String `Daube'>
  <Font
    <Ftag ``'>
    <FLocked No>
  > # end of Font
  <String `, 1989 #6}, {LaBont\x8e , 1989 #11}'>
> # end of ParaLine
```

To

```
<ParaLine
  <String `tatum zrril delenit augue dui ..... nulla facilis {#14}, {#6}, {11}'>
> # end of ParaLine
```

Note: *This may cause overlengthy lines (incorrect formatting) and/or illegal mif. However, FrameMker 7 does not report any MIF errors for this.*

Highlight crosses entry

Input

Tia non ob ea soluad incommod **quae egen ium improb fugiend {Bartels, 1981 #14}, {Daube, 1989 #6}, {LaBonté, 1989 #11}.**

From

```
<ParaLine
  <String `Tia non ob ea soluad incommod '>
  <Font
    <FTag `name'>
    <FLocked No>
  > # end of Font
  <String `quae egen ium improb fugiend {Bar'>
  <Char SoftHyphen>
  > # end of ParaLine
<ParaLine
  <String `tels, 1981 #14}, {Daube, 1989 #6}, {LaBont\x8e , '>
  <Font
    <FTag ` '>
    <FLocked No>
  > # end of Font
  <String `1989 #11}.'>
  > # end of ParaLine
```

To

```
<ParaLine
  <String `Tia non ob ea soluad incommod '>
  <Font
    <FTag `name'>
    <FLocked No>
  > # end of Font
  <String `quae egen ium improb fugiend {#14}, { #6}, {#11}.'>
  > # end of ParaLine
```

Note: *This may create unwanted formatting, since the switch to character format name is not terminated properly. The applied character format is that of the first opening brace.*

Split entry, highlighted entry, multiple entries

Input

tatum zzril delenit augue dui dolore te feugait nulla facilis
 {Bartels, 1981 #14}, {Daube, 1989 #6}, {LaBonté, 1989 #11}

From

```
<ParaLine
  <String `tatum zzril delenit augue dui dolore te feugait nulla facilis {Bar}'>
  <Char SoftHyphen>
> # end of ParaLine
<ParaLine
  <String `tels, 1981 #14}, '>
  <Font
    <FTag `1-highlight'>
    <FLocked No>
  > # end of Font
  <String `{Daube, 1989 #6}'>
  <Font
    <FTag ` '>
    <FLocked No>
  > # end of Font
  <String `, {LaBont\x8e , 1989 #11}'>
> # end of ParaLine
```

To

```
<ParaLine
  <String `tatum zzril delenit augue dui dolore te feugait nulla facilis {#14}, '>
  <Font
    <FTag `1-highlight'>
    <FLocked No>
  > # end of Font
  <String `{#6}'>
  <Font
    <FTag ` '>
    <FLocked No>
  > # end of Font
  <String `, {#11}'>
> # end of ParaLine
```

2010-04-29

E:\FM-specials\EndNote5\Fix_umlaut_problem\fm2endnote5.fm

Code of fm2endnote5

```
; WinBatch 5.0
;FILE:      F:\WinBatch\BATCHE\fm2endnote5.wbt
;PURPOSE:   Modify MIF for use in EndNote 5
;USAGE:     Execute the script; This has been compiled also
;AUTHOR:    Klaus Daube, Docu + Design Daube, Zürich
;VERSION:   1.0
;CREATION:  2003-09-04
;UPDATED:   => history at end of this file
;-----
lookfor =    "<String ""
skiplen =    StrLen (lookfor)
termbegin =  "{"
termend =    "}"
filename=    "*.mif"
extension =  ".mif"
sourceFile=  ""           ;build up complete file name

; --- get next inputline containing a "<String ""
#DefineFunction udfShiftLineL (instring, pos)
; shift contents of line before/including pos1 to the left
    len = StrLen (instring) - pos
    line = StrSub (instring, pos+1, len)
    Return line
#EndFunction

#DefineFunction udfStrSub (instring, pos1, pos2)
    len = pos2 - pos1 + 1
    str = StrSub (instring, pos1, len)
    Return str
#EndFunction

#DefineSubRoutine udfGetLine ()
; using fSource, fOut, lineIn, nLines, lookFor, lineLen
; copy lines not containing the lookFor string to output
    While lineIn <> "**EOF*"
        lineIn = FileRead(fSource)
        nLines = nLines + 1
        If nLines mod 1000 == 0 Then Display (1, "fm2endnote5 - long MIF file", StrCat
("Processing Line ", nLines))
        if StrIndex (lineIn, lookFor, 1, @FWDSCAN) > 0 then
            lineLen = StrLen (lineIn)
            Return
        Else
            If lineIn <> "**EOF*" Then FileWrite (fOut, lineIn)
        EndIf
    EndWhile
    Return
#EndSubRoutine

#DefineSubRoutine udfSkipLines ()
; using fSource, lineIn, nLines, lookFor, lineLen
; skip any lines not containing the lookFor string and remove this from found line
; in this routine an EOF should not be encountered - or input MIF is wrong
    While lineIn <> "**EOF*"
        lineIn = FileRead(fSource)
        nLines = nLines + 1
        If nLines mod 1000 == 0 Then Display (1, "fm2endnote5 - long MIF file", StrCat
("Processing Line ", nLines))
```

```

zpos = StrIndex (lineIn, lookFor, 1, @FWDSCAN)
  if zpos > 0 then
    lineIn = udfShiftLineL (lineIn, skipLen+zpos-1)
    lineLen = StrLen (lineIn)
    Return
  EndIf
EndWhile
Return
#EndSubRoutine

;--- Dialogue to get the necessary information
:reshow
zFormat='WWDLGED,6.1'

zCaption='fm2endnote5: modify MIF for EndNote 5 [D+DD]'
zX=028
zY=072
zWidth=174
zHeight=146
zNumControls=004
fzProcedure='DEFAULT'
zFont='DEFAULT'
zTextColor='DEFAULT'
zBackground='DEFAULT,DEFAULT'
zConfig=123

z001='084,127,032,011,PUSHBUTTON,DEFAULT,"OK",1,1,32,DEFAULT,DEFAULT,DEFAULT'
z002='130,127,034,011,PUSHBUTTON,DEFAULT,"Cancel",0,2,DEFAULT,DEFAULT,DEFAULT,DEFAULT'
z003='007,017,157,078,FILELISTBOX,filename,DEFAULT,DEFAULT,3,DEFAULT,DEFAULT,DEFAULT,DEFAULT'
z004='009,006,110,006,STATICTEXT,DEFAULT,"Select original MIF to be
converted",DEFAULT,4,DEFAULT,DEFAULT,DEFAULT,DEFAULT'
; z005='007,108,126,011,CHECKBOX,doFileRemove,"Delete intermediate files after calling
EndNote",1,5,DEFAULT,DEFAULT,DEFAULT,DEFAULT'

zButtonPushed=Dialog("z")

currPath = DirGet ()
sourceFile = StrCat (currPath, fileName)
sourceName = FileRoot (fileName)
modFile = StrCat (currPath, sourceName, "_mod", extension)
fSource = FileOpen(sourceFile, "READ")
lineIn = FileRead(fSource) ; first input line is header
DirChange (currPath)

fOut = FileOpen(modFile, "WRITE")
nLines = 0

;--- output modified header line
lineOut= "<MIFFile 6.00> # FM7 mif modified by fm2endnote5 [D+DD]"
nLines = nLines + 1

while lineIn <> "*EOF*"
  FileWrite(fOut, lineOut)
; Message ("OutputLine", lineOut) ;-----debug
  lineOut = "" ; for copy/concatination
  udfGetLine () ; get line with string
  If lineIn == "*EOF*" then Break ; bail out
  While lineIn <> ""
    posBegin = StrIndex (lineIn, termBegin, 1, @FWDSCAN) ; a citation present ?
    If posBegin == 0 Then ; no citation here

```

2010-04-29

E:\FM-specials\EndNote5\Fix_umlaut_problem\fm2endnote5.fm

D+D

```

LineOut = StrCat (LineOut, LineIn) ; complete this line
lineIn = "" ; to terminate inner while
Continue ; ... with next input line
Else ; a citation starts on this line
lineOut = StrCat(lineOut, udfStrSub (lineIn, 1, posBegin)) ; begin ... {
posBegin = StrIndexWild(lineIn, "{*, * #*}", 1) ; citation format: {Garneau,
1990 #12}
If posBegin <> 0 Then ; a complete citation present
posEnd = StrIndex (lineIn, termEnd, 1, @FWDSCAN)
posNumber = StrIndex (lineIn, "#", 1, @FWDSCAN)
lineOut = StrCat (lineOut, udfStrSub (lineIn, posNumber, posEnd))
lineIn = udfShiftLineL (lineIn, posEnd) ; for rescan
Else ; either a non-citation {construct} or only start of
citation
udfSkipLines () ; find next relevant line
posNumber = StrIndex (lineIn, "#", 1, @FWDSCAN)
If posNumber <> 0 Then
posEnd = StrIndex (lineIn, termEnd, 1, @FWDSCAN) ; terminating brace
lineOut = StrCat (lineOut, udfStrSub (lineIn, posNumber, posEnd))
lineIn = udfShiftLineL (lineIn, posEnd) ; for rescan
Else ; intermediate lines
While posNumber == 0
udfSkipLines () ; find next relevant line
posNumber = StrIndex (lineIn, "#", 1, @FWDSCAN)
EndWhile
posEnd = StrIndex (lineIn, termEnd, 1, @FWDSCAN) ; terminating brace
lineOut = StrCat (lineOut, udfStrSub (lineIn, posNumber, posEnd))
lineIn = udfShiftLineL (lineIn, posEnd) ; for rescan
Endif
EndIf
EndIf
EndWhile
If lineIn == "*EOF*" then Break ; bail out
EndWhile

;--- Terminate the files properly

FileClose(fOut)
FileClose(fSource)
Message("fm2endnote5", StrCat(modFile,@CR, " generated from ", nLines, " input lines"))

;--- rename/delete intermediate files if EndNote is open and can access the final form

Exit

```