```
//*************************************************************
//* Date:       4/24/95 5:00 pm
//*
//* Macro:      UTRLTXT2.WCM
//*
//* Purpose:    To prepare Right-to-Left text for editing.
//*
//*             The user of this macro is working with a pseudo
//*             paragraph of Right-to-Left text.  A pseudo
//*             paragraph is defined such that each line ends with
//*             a carriage return and the paragraph ends with two
//*             carriage returns.  The cursor can be anywhere inside
//*             the pseudo paragraph when the macro is invoked.  The
//*             lines of the paragraph are reversed such that the
//*             first line becomes the last line and the last line
//*             becomes the first line.  Then the paragraph is stripped
//*             of its carriage returns.
//*
//*             The resulting text is now "Upside Down" but it is
//*             in an order that inserting text or changing margins
//*             will wordwrap and preserve the proper order of the
//*             Right-to-Left text.
//*
//*             When the editing is finished the user will run the
//*             UTRLTXT2 macro which will re-order the text back.
//*
//* Notes:      If the text is not a correct pseudo paragraph the
//*             macro will issue an error MessageBox and return
//*             having done nothing!
//*************************************************************
//* Application (A1; "WordPerfect"; Default; "US")
USE( "UTRLTXT1.WCM")
Display(State: Off!)
WaitMessage()
CodesRevealed:= ?RevealCodesActive
If(CodesRevealed = True)
 RevealCodes(Off!)
EndIf
SwapWhiteSpaceFirstLast()
ConvertTruePghToPseudoPgh()
ReverseLinesOfPseudoPgh()
RemoveWhiteSpaceLast()
GotoStartOfPseudoParagraph()
If(CodesRevealed = True)
 RevealCodes(On!)
EndIf
```

```
KillWaitMessage()
Quit
//=====================================================================
// SwapWhiteSpaceFirstLast()
//
//    This function assumes true paragraphs where all lines
//    excluding the last line ends with a word wrap and the
//    last line ends with a carriage return.
//
//    This function essentially moves the white space from the
//    end of the last line of a true paragraph and puts it on
//    the start of the first line of that paragraph.
//
Function SwapWhiteSpaceFirstLast()
 spaces = 0
 lines =  CountLinesInTrueParagraph()
 toInsert = 32
 While (toInsert >= 1)
   // this while loop inserts toInsert spaces
   // at a time until the line count bumps
   tLines = lines
   While(tLines = lines)
     GotoStartOfTrueParagraph()
     For(i;1;i<=toInsert;i+1)
       Type(" ")
     EndFor
     spaces = spaces + toInsert
     MacroStatusPrompt(On!; "          Calculating: "+spaces)
     tLines = CountLinesInTrueParagraph()
   EndWhile
   // oops too much space so we remove the last insert
   GotoStartOfTrueParagraph
   For(i;1;i<=toInsert;i+1)
     DeleteCharNext()
   EndFor
   spaces = spaces - toInsert
   // now set the insert amount to the next smaller increment
   If (toInsert = 2)
     toInsert = 1
   Else
     toInsert = toInsert/4
   EndIf
 EndWhile
 GotoStartOfTrueParagraph()
EndFunc
//=====================================================================
```

```
// RemoveWhiteSpaceLast()
//
//    This function assumes pseudo paragraphs where each line
//    ends with a single return and each paragraph ends with
//    a double return.
//
//    Because the pseudo paragraph was created from a true
//    paragraph that had been kludged by a call to the
//    SwapWhiteSpaceFirstLast() function we now have unwanted
//    white space on the last line.  This function removes that
//    white space.
//
Function RemoveWhiteSpaceLast()
 lines = CountLinesInPseudoParagraph() - 1
 While(lines <> 0)
   PosLineDown
   lines = lines - 1
 EndWhile
 PosLineBeg
 spaces = 1
 While(?RightChar = " ")
   MacroStatusPrompt(On!; "          Deleting spaces. "+spaces)
   DeleteCharNext
   spaces = spaces + 1
 EndWhile
 Type(" ")
EndFunc
//=========================================================================
// ConvertTruePghToPseudoPgh()
//
//    This function assumes true paragraphs where all lines
//    excluding the last line ends with a word wrap and the
//    last line ends with a carriage return
//
//    This function converts the true paragraph which contains
//    the cursor (or start of selection) to a pseudo paragraph
//    by adding a return to the end of each line excluding the
//    last line.
//
//    The cursor is then positioned at the start of the pseudo
//    paragraph on exit.
//
Function ConvertTruePghToPseudoPgh()
 ForceDoubleReturnOnTruePgh()
 lines = CountLinesInTrueParagraph()
 AddReturnsToLines(lines)
```

```
 GotoStartOfPseudoParagraph()
EndFunc
//=====================================================================
// CountLinesInTrueParagraph()
//
//    This function assumes true paragraphs where all lines
//    excluding the last line ends with a word wrap and the
//    last line ends with a carriage return
//
//    This function counts the number of lines in a true
//    paragraph.
//
//    As a side effect the cursor is left positioned at
//    the start of the paragraph.
//
Function CountLinesInTrueParagraph()
 GotoStartOfTrueParagraph()
 lines = 1
 While(LineType()=1)
   lines = lines + 1
   MacroStatusPrompt(On!; "          Counting lines: "+lines)
   PosLineDown
 EndWhile
 GotoStartOfTrueParagraph()
 return(lines)
EndFunc
//=====================================================================
// ForceDoubleReturnOnTruePgh()
//
//    This function assumes true paragraphs where all lines
//    excluding the last line ends with a word wrap and the
//    last line ends with a carriage return
//
//    This function looks at the end of a true paragraph and
//    forces it to end with 2 carriage returns.
//
Function ForceDoubleReturnOnTruePgh()
 // force preceeding CR Only
 GotoStartOfTrueParagraph()
 PosLineVeryBeg
 If(?LeftCode<>0)
   PosLineUp
   lt = LineType()
  PosLineDown
   If(lt <> 3)
     PosLineBeg
```

```
    HardReturn
  EndIf
 EndIf
 // force ending CR Only
 While(LineType()=1)
  PosLineDown
 EndWhile
 If(LineType() = 5)
  GotoStartOfTrueParagraph()
  return
 EndIf
 PosLineDown
 If(LineType() <>3)
  PosLineUp
  PosLineEnd
  HardReturn
 EndIf
 PosLineUp
 GotoStartOfTrueParagraph()
EndFunc
//=====================================================================
// GotoStartOfTrueParagraph()
//
//   This function assumes true paragraphs where all lines
//   excluding the last line ends with a word wrap and the
//   last line ends with a carriage return
//
//   On input the cursor is anywhere inside such a paragraph.
//   This function will move the cursor to the first character
//   of such a paragraph.
//
Function GotoStartOfTrueParagraph()
 done = 0
 While(done=0)
  // goto to the start of this line
  // if its the start of the DOC we are done
  PosLineVeryBeg
  If(?LeftCode=0)
   done=1
  Else
   //go up one line
   PosLineUp
   Switch(LineType())
    CaseOf 2 :
     Continue
    CaseOf 3 :
```

```
        PosLineDown
        PosLineBeg
        done = 1
      EndSwitch
    EndIf
  EndWhile
  PosLineBeg
EndFunc
//=====================================================================
// AddReturnsToLines()
//
//    This function assumes true paragraphs where all lines
//    excluding the last line ends with a word wrap and the
//    last line ends with a carriage return
//
//    Returns are added to the passed number of lines starting
//    with the line containing the cursor (or startr of selection).
//
Function AddReturnsToLines(lines)
 While(lines<>0)
   PosLineVeryEnd
   If(?RightCode<>204)
     If(?LeftChar = " ")
       DeleteCharPrevious
     EndIf
     HardReturn()
     // Spaces appear to be stored after the end of wrapped
     // lines so we must delete one if it now shows up
     If(?RightChar = " ")
       DeleteCharNext
     EndIf
   EndIf
   lines = lines - 1
 EndWhile
EndFunc


// Commented for later use:
//    MessageBox(action;"Warning!";"LeftChar ["+?LeftChar+"]  RightChar ["+?
RightChar+"]  //LeftCode "+?LeftCode+"  RightCode "+?RightCode;IconStop! | OK!;)
```