

```

*****
//*   MACRO: FOOTEND.WCM
//*   PURPOSE: Converts footnotes to endnotes. Will convert the entire document or a
selected section.
*****
Application (A1; "WordPerfect"; Default; "US")

If(?DocBlank)
    MessageBox( "Error"; "Document is blank."; IconStop!)
    Quit
EndIf

If(?Substructure)
    MessageBox( "Not At Main Edit Screen"; "This macro can be played only from the main
editing screen. Exit all substructures (headers, text boxes, etc.) and play the macro
again."; IconStop!)
    Quit
EndIf

// set up search conditions
SearchCaseSensitive (No!)
SearchFindWholeWordsOnly (No!)
MatchWithAttributes (No!)
MatchWithFontSize (No!)
MatchWithFont (No!)
MatchPositionAfter()
SearchInSelection (No!)
SearchWrap (No!)

If(?BlockActive<>0)
    Call(Select@) //Convert selected text only.
Else
    Call(Doc@) //Convert entire document.
EndIf

Label(End@)
// Set the search options back to the default
MatchSelection ()
MacroStatusPrompt(Off!)
Quit

*****
//*   ROUTINE: Doc@
//*   INPUT VARIABLES: None
//*   OUTPUT VARIABLES: Select
//*   DESCRIPTION: Converts footnotes to endnotes in the whole document.

```

```

//*****
Label(Doc@)
Select:=False
WaitMessage:="Converting all footnotes to endnotes"
Call(WaitMessage@)
MacroStatusPrompt(On!; "Converting all footnotes to endnotes...")
PosDocBottom()
OnNotFound(NoFootNotes@)
While(True)
    SearchString("[Footnote]")
    SearchPrevious(Extended!)
    FootnoteEdit()
    Call(SwapInfo@)
    OnNotFound(DocDone@)
EndWhile
Label(DocDone@)
Call(KillWaitMessage@)
Return
//*****

//*****
/*    ROUTINE: SwapInfo@
/*    INPUT VARIABLES: Select
/*    OUTPUT VARIABLES: None
/*    DESCRIPTION: Copies text from open footnote to a new endnote.
//*****
Label(SwapInfo@)
Number4Display:=?Footnote
If(Select)
    Number4Display:=Number4Display-1
EndIf
If(Not((?RightCode = 0) And (?RightChar = "")))
    SelectDocBottom()
    EditCopy()
    TextToPaste:=True
Else
    TextToPaste:=False
EndIf
SubstructureExit()
MacroStatusPrompt(On!; "Converting Footnote "+Number4Display)
EndnoteCreate()
If(TextToPaste)
    EditPaste()
EndIf
SubstructureExit()
PosCharPrevious()

```

DeleteCharPrevious()  
Return

```
*****  
/* SUBROUTINE: Select  
/* INPUT VARIABLES: None  
/* OUTPUT VARIABLES: Select  
/* DESCRIPTION: Converts Footnotes to endnotes in selected text.  
*****  
Label(Select@)  
Select:=True  
WaitMessage:="Converting selected footnotes to endnotes"  
Call(WaitMessage@)  
MacroStatusPrompt(On!; "Converting selected footnotes to endnotes...")  
PosSelectTop ()  
SelectOff()  
PosCharPrevious()  
FootNoteCreate()  
MyFootnote:=?Footnote  
SubstructureExit()  
ReselectLastSelection ()  
PosSelectBottom()  
SelectOff()  
Check:=0  
Label(Select1@)  
    Check:=Check+1  
    SearchString("[Footnote]")  
    SearchPrevious(Extended!)  
    FootnoteEdit()  
    If(?Footnote<>MyFootnote)  
        Call(SwapInfo@)  
        OnNotFound(SelectDone@)  
        GO(Select1@)  
    EndIf  
SubstructureExit()  
DeleteCharPrevious()  
If(Check=1)  
    Go(NoFootNotes@)  
EndIf  
Label(SelectDone@)  
Call(KillWaitMessage@)  
Return  
*****  
  
*****  
/* ROUTINE: NoFootNotes@
```

```

/* INPUT VARIABLES: Select
/* OUTPUT VARIABLES: None
/* DESCRIPTION: Displays an error message.
*****
Label(NoFootNotes@)
If(Select)
    MsgBoxMessage:="No footnotes were found in the selected text."
Else
    MsgBoxMessage:="No footnotes were found in the document."
EndIf
MessageBox(; "No Footnotes Found"; MsgBoxMessage; IconStop!)
Go(End@)
*****

*****
/* ROUTINE: WaitMessage@
/* INPUT VARIABLES: WaitMessage
/* OUTPUT VARIABLES: WaitMessageResult
/* DESCRIPTION: Presents a Please Wait message to user. Call KillWaitMessage to turn
it off.
*****
Label(WaitMessage@)
WaitTitle:="WAITPROMPT"
WaitMessageWidth:=138
DialogDefine(WaitTitle; 50; 50; WaitMessageWidth+22; 43; 16+256+8+2; WaitTitle)
RegionShowWindow(WaitTitle+".CancelBtn"; Hide!)
DialogAddText(WaitTitle; "T1"; 8; 8; 46; 10; 1; "Please Wait...")
DialogAddText(WaitTitle; "T2"; 8; 19; WaitMessageWidth; 10; 1; WaitMessage)
DialogDisplay(WaitTitle; 1; WaitDlgCallBack@)
Return

Label(WaitDlgCallBack@)
If(WaitDlgCallBack[3] = "CancelBtn")
    Assert(CancelCondition!)
EndIf
Return
*****

*****
/* ROUTINE: KillWaitMessage@
/* INPUT VARIABLES: none
/* OUTPUT VARIABLES: none
/* DESCRIPTION: Turns off wait message.
*****
Label(KillWaitMessage@)
DialogDestroy(WaitTitle)

```

Return

//\*\*\*\*\*