

```

*****
/**      MACRO: ENDFOOT.WCM
/**      PURPOSE: Converts endnotes to footnotes. 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 endnotes to footnotes in the whole document.

```

```

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

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

```

Return

```

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

/*****
/*   ROUTINE: NoEndNotes@
/*   INPUT VARIABLES: Select
/*   OUTPUT VARIABLES: None

```

```

/** DESCRIPTION: Displays an error message.
*****
Label(NoEndNotes@)
If(Select)
    MsgBoxMessage:="No endnotes were found in the selected text."
Else
    MsgBoxMessage:="No endnotes were found in the document."
EndIf
MessageBox( "No Endnotes 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
*****

```