

```

*****
/*      MACRO: EXPNDALL.WCM
/*      PURPOSE: Expand all abbreviations in the document that are defined in the current,
default and supplemental templates.
*****
Application (A1; "WordPerfect"; Default; "UK")

If(?DocBlank)
    Go(Blank@)
EndIf

Declare(NumPerLoc[3])
NumPerLoc[1]:=0
NumPerLoc[2]:=0
NumPerLoc[3]:=0
WaitMessage:="Retrieving abbreviations from templates"
Call(WaitMessage@)
If(?GraphicSelected)
    BoxEnd(Save!)
EndIf
If(Not ?SubStructure)
    MacroStatusPrompt(On!; "Retrieving abbreviations from templates...")
EndIf
For(Loop; 0; Loop<3; Loop+1)    // 0=Current, 1=Default, 3=Supplemental
    GetData(AbbCnt; Abbreviation!; Count!; Loop)
    NumPerLoc[Loop+1]:=AbbCnt
EndFor
If(?CurrentTemplate=?TemplateFile)
    NumPerLoc[2]:=0    // The current template is the default template, so don't count it
twice.
EndIf
TotalAbbrev:=NumPerLoc[1]+NumPerLoc[2]+NumPerLoc[3]
If(TotalAbbrev=0)
    Go(End@)    // there are no abbreviations
EndIf
Declare(Abbreviation[TotalAbbrev; 2])
TotalCnt:=0
For(Location; 0; Location<3; Location+1)
    For(Loop; 1; Loop<NumPerLoc[Location+1]+1; Loop+1)
        GetData(AbbrevName; Abbreviation!; Name!; Location; Loop)
        // skip automated template and address file information
        If((SubStr(AbbrevName; 1; 4) <> "REC:") And (AbbrevName<>"Personal
Information") And (AbbrevName<>"Template Prompts"))
            TotalCnt:=TotalCnt+1
            Abbreviation[TotalCnt; 1]:=AbbrevName
            Abbreviation[TotalCnt; 2]:=Location
        EndIf
    EndFor
EndFor

```

```

        EndIf
    EndFor
EndFor
// now we have all abbreviation names stored into the array. So let's search for the abbreviations
one at a time and expand the word when we find it.
SearchFindWholeWordsOnly(Yes!)
SearchCaseSensitive(Yes!)
MatchSelection()
SearchInSelection(No!)
SearchWrap(No!)
OnNotFound(NextAbbreviation@)
WaitMessage:="Expanding all abbreviations in document"
Call(KillWaitMessage@)
Call(WaitMessage@)
For(Loop; 1; Loop<=TotalCnt; Loop+1)
    While(?Substructure)
        SubDoc:=?CurrentSubDoc
        SubstructureExit()
        If((SubDoc=10) Or (SubDoc=11))
            BoxEnd(Save!)
        EndIf
    EndWhile
    PosDocVeryTop()
    If(Not ?SubStructure)
        MacroStatusPrompt(On!; "Expanding abbreviation "+Abbreviation[Loop;
        1]+" ...")
    EndIf
    Repeat
        SearchString(Abbreviation[Loop; 1])
        SearchNext(Extended!)
        AbbreviationExpand()
    Until(False)
    Label(NextAbbreviation@)
EndFor
Call(KillWaitMessage@)
    While(?Substructure)
        SubDoc:=?CurrentSubDoc
        SubstructureExit()
        If((SubDoc=10) Or (SubDoc=11))
            BoxEnd(Save!)
        EndIf
    EndWhile
SearchFindWholeWordsOnly(No!)
SearchCaseSensitive(No!)

Label(End@)

```

Quit

```
/**
*****
/**  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:=132
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
*****

/**
*****
/**  ROUTINE: Blank@
/**  INPUT VARIABLES: None
/**  OUTPUT VARIABLES: None
/**  DESCRIPTION: Displays an error message.
*****
Label(Blank@)
MessageBox(; "Error"; "Document is blank."; IconStop!)
Go(End@)
```

//*****