

```

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

If (?DocBlank)
    Go (Blank@)
EndIf

Declare (NumPerLoc[3])
NumPerLoc[1] := 0
NumPerLoc[2] := 0
NumPerLoc[3] := 0
WaitMsgInit ()
WaitMsgDisplay ("Retrieving QuickWords from templates")
If (?GraphicSelected)
    BoxEnd (Save!)
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 QuickWords
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 information
        If ((AbbrevName <>"Template Bookmark Info") And (AbbrevName <>"Template
Prompt Info"))
            TotalCnt := TotalCnt + 1
            Abbreviation[TotalCnt; 1] := AbbrevName
            Abbreviation[TotalCnt; 2] := Location
        EndIf
    EndFor
EndFor
EndFor

```

```

// now we have all QuickWord names stored into the array. So let's search for the QuickWords
one at a time and expand the word when we find it.
SearchFindWholeWordsOnly (Yes!)
SearchCaseSensitive (Yes!)
MatchSelection ()
SearchInSelection (No!)
SearchWrap (No!)
OnNotFound (NextAbbreviation@)
WaitMsgDisplay ("Expanding all QuickWords in document")
For (Loop; 1; Loop <= TotalCnt; Loop + 1)
    While (?Substructure)
        SubDoc := ?CurrentSubDoc
        SubstructureExit ()
        If ((SubDoc = 10) Or (SubDoc = 11))
            BoxEnd (Save!)
        EndIf
    EndWhile
    PosDocVeryTop ()
    WaitMsgDisplay ("Expanding QuickWord " + Abbreviation[Loop; 1] + "...")
    Repeat
        If (?GraphicSelected) // ab: check for graphic objects
            BoxAttachTo(Anchor:Character!) //lv
            BoxEnd (Save!)
        EndIf
        SearchString (Abbreviation[Loop; 1])
        SearchNext (Extended!)
        AbbreviationExpand ()
    Until (False)
    Label (NextAbbreviation@)
EndFor
WaitMsgHide ()
WaitMsgDestroy ()
While (?Substructure)
    SubDoc := ?CurrentSubDoc
    SubstructureExit ()
    If ((SubDoc = 10) Or (SubDoc = 11))
        BoxEnd (Save!)
    EndIf
EndWhile
SearchFindWholeWordsOnly (No!)
SearchCaseSensitive (No!)

Label (End@)
Quit

```

```

//*****

```

```

/*    ROUTINE: Blank@
/*    INPUT VARIABLES: None
/*    OUTPUT VARIABLES: None
/*    DESCRIPTION: Displays an error message.
*****
Label (Blank@)
MessageBox (; "Error"; "There are no QuickWords to expand in this document."; IconStop!)
Go (End@)
*****

*****
//    PROCEDURE: WaitMsgInit ()
//    PURPOSE: Initializes macro wait messages.
*****
PROCEDURE WaitMsgInit ()
DialogDefine ("WaitMsg"; 50; 50; 150; 36; 16; "Please Wait")
DialogAddText ("WaitMsg"; "WaitText"; 8; 14; 144; 16; 1; "")
DialogLoad ("WaitMsg")
Return
ENDPROC

*****
//    PROCEDURE: WaitMsgDisplay (Text)
//    PURPOSE: Displays a previously defined wait message dialog.
*****
PROCEDURE WaitMsgDisplay (Text)
RegionSetWindowText ("WaitMsg" + ".WaitText"; Text)
DialogShow ("WaitMsg";; WaitDlgCallBack)
ENDPROC

PROCEDURE WaitDlgCallBack ()
If (WaitDlgCallBack[5] = 274 And WaitDlgCallBack[6] = 61536)
    Assert (CancelCondition!)
EndIf
ENDPROC

*****
//    PROCEDURE: WaitMsgHide
//    PURPOSE: Hides a previously defined wait message dialog.
*****
PROCEDURE WaitMsgHide ()
DialogDismiss ("WaitMsg"; "WaitText")
ENDPROC

*****
//    PROCEDURE: WaitMsgDestroy

```

```
//      PURPOSE: Destroys a previously defined wait message dialog.  
//*****  
PROCEDURE WaitMsgDestroy ()  
DialogDestroy ("WaitMsg")  
ENDPROC  
//*****
```