```
//*************************************************************
//*       MACRO: ALLFONTS.WCM
//*       PURPOSE: Creates a list of all font faces available.
//*************************************************************
Application (A1; "WordPerfect"; Default; "EN")
If (Not ?DocBlank)
        If( ?NumberOpenDocuments < 9)
                FileNew()
        Else
                MessageBox(; "Error!"; "You must close at least one document before you can
                        run this macro")
                Go( End@)
        EndIf
EndIf
If (?SubStructure)
        MessageBox (; "Error - Not In Main Edit Screen"; "This macro can only be run from the
        main editing screen."; IconStop!)
        Go (End@)
EndIf
WaitMsgInit ()
WaitMsgDisplay ("Retrieving font information")
BaseFont := ?Font
GetData (FontCnt; Font!; Count!; CurrentDoc!)
Declare (FontInfo[FontCnt])
FontCount := 0
For (Count; 1; Count <= FontCnt; Count + 1)
        GetData (FontName; Font!; Name!; CurrentDoc!; Count)
        If (SubStr (FontName; 1; 2) <>"WP")
                FontCount := FontCount + 1
                FontInfo[FontCount] := FontName
        EndIf
EndFor
For (Count; 1; Count <= FontCount; Count + 1)
        WaitMsgDisplay ("Inserting " + FontInfo[Count] + "...")
        BlockProtect (On!)
        AttributeAppearanceOn (Underline!)
        Type (FontInfo[Count])
        AttributeAppearanceOff (Underline!)

        HardReturn ()
        Font (FontInfo[Count])
        Type
        ("aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ1234567890`'"
        "~!@#$%^&*()_+-=:;,.?/\|<>[]{}")
        Font (BaseFont)
        BlockProtect (Off!)
```

```
        HardReturn ()
        HardReturn ()
EndFor

SortKeys (1; 1; 1; Alphanumeric!; Ascending!)
SortType (ParagraphSort!)
SortAction (Sort!)
SortCaseOrder (LowercaseFirst!)
Sort ()
PosDocVeryTop ()
Type ("You have " + FontCount + " fonts available to the " + ?CurrentPrinter + ":")
HardReturn ()
WaitMsgHide ()
WaitMsgDestroy ()

Label (End@)
Quit

//****************************************
//      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
//*********************************************************
```