

```

//*****
//*   MACRO: ALLFONTS.WCM
//*   PURPOSE: Creates a list of all font faces available.
//*****
Application (A1; "WordPerfect"; Default; "UK")
OnError(End@)
OnCancel(End@)
Display(Off!)
If(Not ?DocBlank)
    MessageBox(; "Document Is Not Blank"; "The font list must be placed in a blank
document."; IconStop!)
    Go(End@)
EndIf
WaitMessage:="Retrieving font information"
Call(WaitMessage@)
MacroStatusPrompt(On!; "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
WaitMessage:="Placing sample text in document"
Call(KillWaitMessage@)
Call(WaitMessage@)
NumStr(FontCountStr; 0; FontCount)
For(Count; 1; Count<=FontCount; Count+1)
    MacroStatusPrompt(On!; "Placing sample text for "+FontInfo[Count]+" ...")
    BlockProtect(On!)
    AttributeAppearanceOn(Underline!)
    Font(BaseFont)
    Type(FontInfo[Count])
    AttributeAppearanceOff(Underline!)

    HardReturn()
    Font(FontInfo[Count])
    Type("aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ12345678
90`~!@#$%^&*()_+ -=:;,./\|<>[]{} ")
    BlockProtect(Off!)
    HardReturn()
    HardReturn()

```

EndFor

```
MacroStatusPrompt(On!; "Sorting font list...")
SortKeys(1; 1; 1; Alphanumeric!; Ascending!)
SortType(ParagraphSort!)
SortAction(Sort!)
SortCaseOrder(LowercaseFirst!)
Sort()
PosDocVeryTop()
Type("You have "+FontCountStr+" fonts available to the "+?CurrentPrinter+":")
HardReturn()
```

```
Label(End@)
MacroStatusPrompt(Off!)
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:=105
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

//*****