

```

// This macro closes and then reopens all the current
// documents, which causes WordPerfect to rebuild its
// list of available fonts.
//
APPLICATION(A1; "WordPerfect"; Default; "US")

// Display a message if the user cancels.
//
OnCancel(UserCancel@)

// Count the number of open documents. Is there an easier
// way to do this?
//
wStartDoc := ?DocNumber
wNumDocs := 0
REPEAT
    DocNext()
    wNumDocs := wNumDocs + 1
UNTIL (?DocNumber = wStartDoc)

// If the blank startup doc is visible, with no changes
// from the user, then just update its list without
// prompting.
//
IF ((wNumDocs = 1) AND (?Name = "") AND ?DocBlank)
    CloseNoSave()
    Quit
ENDIF

// Otherwise, instruct the user that the font list is
// now out of sync and offer to update it for them.
//
MsgBoxMessage := "You used Font Works to change application font menus. Choose
'OK' to update WordPerfect's font menu or choose 'Cancel' to leave the menu
unchanged. WordPerfect will close and reopen all open documents, prompting you to
save any unsaved changes."
MsgBoxCaption := "Update font menu?"
MsgBoxFlags := 49 // okcancel + iconexclam
Call(MessageBox@)
IF (MsgBoxResult = 2) // Cancel
    Quit
ENDIF

// For each doc.
//
FOR (i; 0; i < wNumDocs; i + 1)

```

```

// Get the name.
//
szName := ?Name

// Kill any selected text because cancelling from the
// 'save whole thing or only selected text' dialog does
// not cancel the macro, and we end up opening up a
// second, read-only version, of the doc.
//
SelectOff()

// For a new document, prompt the user to save it. If the
// user cancels from the prompt, then the macro exits.
//
IF ((szName = "") AND NOT ?DocBlank)
    MsgBoxMessage := "Save changes to Document" + ?DocNumber + "?"
    MsgBoxCaption := "WordPerfect for Windows"
    MsgBoxFlags := 35 // yesnocancel + iconquestion
    Call(MessageBox@)
    IF (MsgBoxResult = 6) // Yes
        FileSaveAsDlg()
        szName := ?Name
    ENDIF
    IF (MsgBoxResult = 2) // Cancel
        Assert(CancelCondition!)
    ENDIF
ENDIF

// Given we know the name of the file (so we can reopen
// it), close and reopen it.
//
IF (szName <> "")

    // Update status.
    //
    szPath := ?Path + szName
    szStat := "Closing " + szPath
    MacroStatusPrompt(On!; szStat)

    // Close the document. If it is dirty, close with a
    // prompt to save. If the user selects cancel, then
    // the macro exits.
    //
    IF (?DocChanged AND NOT ?DocReadOnly)

```

```

    MsgBoxMessage := "Save changes to Document" + szPath + "?"
    MsgBoxCaption := "WordPerfect for Windows"
    MsgBoxFlags := 35 // yesnocancel + iconquestion
    Call(MessageBox@)
    IF (MsgBoxResult = 6) // Yes
        Close(Yes!)
    ELSE
        IF (MsgBoxResult = 7) // No
            Close(No!)
        ELSE // Cancel
            Assert(CancelCondition!)
        ENDIF
    ENDIF
ELSE
    CloseNoSave()
ENDIF

// Update status.
//
szStat := "Opening " + szPath
MacroStatusPrompt(On!; szStat)

// Reopen the file.
//
FileOpen(szPath)

// Change page to the next document.
//
DocNext()

ELSE

    // The document has no name and has recieved no text,
    // so just close it without reopening.
    //
    szStat := "Closing Document" + ?DocNumber
    MacroStatusPrompt(On!; szStat)
    CloseNoSave()

ENDIF
ENDFOR

// Normal exit.

```

```

//
MacroStatusPrompt(Off!)
Quit

// User cancel.
//
Label(UserCancel@)
Cancel(Off!)
MacroStatusPrompt(Off!)
MsgBoxMessage := "In order to update WordPerfect's font menu, you must close and
re-open all open documents or exit and restart WordPerfect."
MsgBoxCaption := "Cancel font menu update"
MsgBoxFlags := 48 // ok + iconexclam
Call(MessageBox@)
Quit

//*****
/*    ROUTINE: MessageBox@
/*    INPUT VARIABLES: MsgBoxCaption; MsgBoxMessage; MsgBoxFlags
/*    OUTPUT VARIABLES: MsgBoxResult
/*    DESCRIPTION: Displays a message box.
//*****
Label(MessageBox@)
Cancel(Off!)
DllLoad(MsgBoxLink; "shwin20.dll")
Display(On!)
DllCall(MsgBoxLink; "WmbMsgBox"; MsgBoxResult:WORD;
        {LoWord(0); LoWord(0); AnsiString(MsgBoxCaption);
        AnsiString(MsgBoxMessage); MsgBoxFlags; 0; LoWord(0)})
Display(Off!)
DllFree(MsgBoxLink)
Cancel(On!)
Return

```