```
//*****************************************************
//Copyright (C) 1997 Maximizer Technologies Inc.
//*****************************************************
//SendDocument
Application (A1; "WordPerfect"; Default; "US")

//declare constants
global mwCurrent; mwMarked; mwEntire; stPrinter; stEMail

//declare shared variables
global ddeCurrent; MassMailInitialized; DLLInitialized; hDLL
global AutoExitPrintProcess; lastEntry; currentEntry
global AttemptToLogNote; NoteAlreadyTested; eMailExist
global hActiveHwnd

//initialize constants
mwCurrent = 0
mwMarked = 1
mwEntire = 2
stPrinter = 0
stEMail = 1
eMailExist = 1

//initialize global variables
DLLInitialized = false
ddeCurrent = 0
hActiveHwnd = AppLocate("Corel WordPerfect*")

onError(Exit_SendDocument)

//initialize local variables
mergeWith = 0
sendTo = 0
note = ""

if ( not MaximizerIsLoaded() ) go Exit_SendDocument endif
if ( not DatabaseIsOpen() ) go Exit_SendDocument endif
if ( not LoopUntilMaximizerIsReady() ) go Exit_SendDocument endif
if ( ListIsEmpty() ) go Exit_SendDocument endif
if ( PersonalHotList() ) go Exit_SendDocument endif

if (DisplayDialog( &mergeWith; &sendTo; &note ) )
      DoSend( mergeWith; sendTo; note )
endif

Label(Exit_SendDocument)
quit


//*****************************************************
//displays the dialog & returns the selections; returns
//false on cancel
//*****************************************************
function DisplayDialog( &mergeWith; &sendTo; &note)
      onError( Exit_DisplayDialog )
      WM_SYSCOMMAND = 274
      WM_COMMAND = 273
```

```
      dlgResult = 999
      rNote = ?Name
      if (rNote = "")
            rNote = "Document" + numstr(?DocNumber)
      endif
rSend = 1
rMerge=1
DialogDefine (Dialog:"SendDocument"; Left:50; Top:50; Width:177; Height:120;
Style:Percent!; Caption:"Send Document")
DialogSetProperties (Dialog:"SendDocument"; FontName:"MS Sans Serif";
FontSize:8p)
DialogAddGroupBox (Dialog:"SendDocument"; Control:"grpMerge"; Left:5; Top:5;
Width:112; Height:43; Title:"Merge with:")
DialogAddRadioButton (Dialog:"SendDocument"; Control:"radCurrent"; Left:10;
Top:15; Width:55; Height:10; ButtonText:"&Current Entry"; MacroVar:rMerge)
DialogAddRadioButton (Dialog:"SendDocument"; Control:"radMarked"; Left:10;
Top:25; Width:56; Height:10; ButtonText:"&Selected List"; MacroVar:var2)
DialogAddRadioButton (Dialog:"SendDocument"; Control:"radEntire"; Left:10;
Top:35; Width:52; Height:10; ButtonText:"&Entire List"; MacroVar:var3)
DialogAddGroupBox (Dialog:"SendDocument"; Control:"grpSend"; Left:5; Top:51;
Width:112; Height:33; Title:"Send document to:")
DialogAddRadioButton (Dialog:"SendDocument"; Control:"radPrinter"; Left:10;
Top:60; Width:54; Height:10; ButtonText:"&Printer"; MacroVar:rSend;
Style:RadioAuto!)
DialogAddRadioButton (Dialog:"SendDocument"; Control:"radEMail"; Left:10;
Top:72; Width:54; Height:10; ButtonText:"E-&Mail"; MacroVar:var6;
Style:RadioAuto!)
DialogAddGroupBox (Dialog:"SendDocument"; Control:"grpNote"; Left:5; Top:86;
Width:162; Height:27; Title:"Message to Log to &Notes:")
DialogAddEditBox (Dialog:"SendDocument"; Control:"txtNote"; Left:11; Top:96;
Width:150; Height:14; Style:Left!+AutoHScroll!; MacroVar:rNote; LimitText:100)
DialogAddPushButton (Dialog:"SendDocument"; Control:"btnOK"; Left:125; Top:9;
Width:42; Height:14; Style:OKBttn!; ButtonText:"OK")
DialogAddPushButton (Dialog:"SendDocument"; Control:"btnCancel"; Left:125;
Top:28; Width:42; Height:14; Style:CancelBttn!; ButtonText:"Cancel")
DialogAddPushButton (Dialog:"SendDocument"; Control:"btnHelp"; Left:126;
Top:48; Width:42; Height:14; Style:NonDefaultBttn!; ButtonText:"Help")

      DialogShow("SendDocument"; "WordPerfect"; HandleDialog; "radCurrent")
      while( dlgResult = 999)
      endWhile
      DialogDismiss("SendDocument"; "btnOK")
      go( Exit_DisplayDialog )

      Label(HandleDialog)
      if (HandleDialog[5] = WM_COMMAND)
            if (HandleDialog[3] = "btnOK")
                  dlgResult = -1
            endif
            if (HandleDialog[3] = "btnCancel") dlgResult = 0 endif
            if (HandleDialog[3] = "btnHelp") DisplayHelp(58001) endif
            if (HandleDialog[3] = "radEMail")
                  sendTo = stEMail
                  //RegionEnableWindow( "SendDocument.txtNote"; 0)
            endif
            if (HandleDialog[3] = "radPrinter")
                  sendTo = stPrinter
```

```
                  //RegionEnableWindow( "SendDocument.txtNote"; 1)
            endif
            if (HandleDialog[3] = "radCurrent") mergeWith = mwCurrent endif
            if (HandleDialog[3] = "radMarked") mergeWith = mwMarked endif
            if (HandleDialog[3] = "radEntire") mergeWith = mwEntire endif
      endif

      if (HandleDialog[5] = WM_SYSCOMMAND) dlgResult = 0 endif
      return

      Label( Exit_DisplayDialog )
      note = rNote
      return ( dlgResult = -1 )
endfunc


//*****************************************************************
//prints the current document for each entry in the Maximizer list
//*****************************************************************
procedure DoSend(mergeWith; sendTo; note )
      AttemptToLogNote = true
      NoteAlreadyTested = false
      if (not LoopUntilMaximizerIsReady() ) go Exit_DoSend endif

      ddeCurrent = InitiateMaxWinDDE("CURRENTRECORD")
      if ( ddeCurrent = 0 ) go Exit_DoSend endif
      onError( DoDDETerminate )

      selectedItemsExist = false
      if (MoveToFirstEntry( mergeWith ) )
            selectedItemsExist = (mergeWith = mwMarked)
            if ( sendTo = stEMail )
                  MassMailInitialized = false
                  DLLInitialized = false

                  if ( not SaveFileIfNecessary() ) Go Exit_DoSend endif
                  path = ?path
                  disp = ?name
                  filename = path
                  if ( substr( fileName; strlen(filename); 1) <> "\")
                        fileName = fileName + "\"
                  endif
                  fileName = fileName + disp

            endif

            if ( sendTo = stPrinter) SetPrintSpooler endif
            onError( DoResetSpooler)

            repeat
                  if ( sendTo = stPrinter) success = SendToPrinter() endif
                  if ( sendTo = stEMail)
                        success = SendToEMail( mergeWith; &note; filename;
path; disp)
                  endif

                  if ( not success ) go DoResetSpooler endif
```

```
                  if ( sendTo = stPrinter) WaitForPrintProcessToFinish endif
                  if ( note <> "" )
                        if (not LogNote( note; sendTo ) ) go DoResetSpooler
endif
                  endif
            until (not MoveToNextEntry( mergeWith ) )

            Label( DoResetSpooler )
            if ( sendTo = stPrinter) ResetPrintSpooler endif
            if ( sendTo = stEMail)
                  if (MassMailInitialized) TerminateMassMail endif
                  if (DLLInitialized) RemoveDLL endif
            endif
      endif

      Label( DoDDETerminate )
      ddeTerminate( ddeCurrent )
      ddeCurrent = 0

      Label( Exit_DoSend)
      if ( ( not selectedItemsExist) and ( mergeWith = mwMarked) )
            messageBox( vDlg; "Maximizer Template"; "You do not have a current
Address Book entry. Please select a company/individual in Maximizer."; OK! |
IconExclamation!)
      endif
endproc


//*************************************************************
//print the current document, return false if it fails
//*************************************************************
function SendToPrinter()
      retValue= false
      onError(Exit_SendToPrinter)

      PrintFullDoc
      retValue = true

      Label(Exit_SendToPrinter)
      return( retValue )
endfunc



//*************************************************************
//EMails the document, returns false if it fails
//*************************************************************
function SendToEMail( mergeWith; &note; filename; path; disp)
      retValue = false
      DDEUpdateLink()
      FileSave( FileName: fileName; OverWrite: YES! )

      onError(Exit_SendToEMail)

      if (not DLLInitialized)
            hDLL = 0
```

```
            dllLoad( hDLL; "MXMAIL97.DLL")
            if ( hDLL < 32 ) go Exit_SendToEMail endif
            DLLInitialized = true
      endif

      //initialize a mass mail if necessary
      if ((mergeWith <> mwCurrent ) and (not MassMailInitialized))
            dllCall( hDLL; "_mxInitializeMassMail@24"; r: integer;
{ hActiveHwnd; 0; address(ansistring(note)); ansistring(fileName);
ansistring(path); ansistring(disp) })
            if ( r <> 0) go Exit_SendToEMail endif
            MassMailInitialized = true
      endif

      email = ddeRequest(ddeCurrent; "MXZ_NEW_EMAIL_FIELD")
      if (email = "")
            eMailExist = 0
      else
            eMailExist = 1
      endif

      if ( mergeWith = mwCurrent )
            if (eMailExist = 0)
                  messagebox( v; "Maximizer Template"; "The current
company/individual does not have an E-mail address."; OK! | IconExclamation!)
                  go Exit_SendToEMail
            endif

//          dllCall( hDLL; "_mxSendSingleMail@24"; r: integer; { hActiveHwnd;
ansistring(email); address(ansistring(note)); ansistring(fileName);
ansistring(path); ansistring(disp) })

            dllCall( hDLL; "_mxSendSingleMail@24"; r: integer; { hActiveHwnd;
ansistring(email); address(ansistring(note)); ansistring(fileName);
ansistring(path); ansistring(disp) })

            if (r = 0)
                  retValue = true
            endif
      else
            newFax = ddeRequest( ddeCurrent; "MXZ_NEW_FAX_FIELD")
            oldFax = ddeRequest( ddeCurrent; "Fax_Number")

            dllCall( hDLL; "_mxSendMassMail@16"; r: integer; { hActiveHwnd;
ansistring(email); ansistring(newfax); ansistring(oldfax) } )
            if (r = 0)
                  retValue = true
            endif
      endif

      Label(Exit_SendToEMail)
      return( retValue )
endfunc


//*************************************************************
//complete the mass mail
```

```
//*****************************************************************
procedure TerminateMassMail()
      dllCall( hDLL; "_mxTerminateMassMail@4"; r: integer; { hActiveHwnd})
      MassMailInitialized = false
endproc


//*****************************************************************
//unload the DLL
//*****************************************************************
procedure RemoveDLL()
      dllFree( hDLL)
      DLLInitialized = false
endproc


//*****************************************************************
//returns true on success, false if DDE failed or if there are no
//entries in the list
//*****************************************************************
function MoveToFirstEntry( mergeWith )
      retValue = false

      numberOfEntries = ListCount()
      if ( numberOfEntries = 0) go Exit_MoveToFirstEntry endif

      onError( Exit_MoveToFirstEntry )
      switch( mergeWith )
            caseof mwCurrent:
                  retValue = true
            caseof mwMarked:
                  ddeExecute( ddeCurrent; "[ScrollList(Top)]" )
                  if (not IsMarked() )
                        onError( Exit_MoveToFirstEntry )
                        ddeExecute( ddeCurrent; "[ScrollList(MarkDown)]" )
                        retValue = IsMarked()
                        onError( Exit_MoveToFirstEntry )
                  else
                        retValue = true
                  endif
            caseof mwEntire:
                  ddeExecute( ddeCurrent; "[ScrollList(Top)]" )
                  retValue = true
      endswitch

      lastEntry = strNum( ddeRequest( ddeCurrent; "GetListEntryIndex" ) )
      if ( ddeRequest( ddeCurrent; "Identification" ) = "" )
            retValue = MoveToNextEntry(mergeWith)
      endif

      Label( Exit_MoveToFirstEntry )
      return( retValue )
endfunc


//*****************************************************************
//moves to the next entry in the current controlling window; skips
```

```
//past any personal entries in the hotlist. Returns true on success
//*****************************************************************
function MoveToNextEntry( mergeWith )
      retValue = false
      onError( Exit_MoveToNextEntry )

      entryValid = false
      while (not entryValid)

            switch( mergeWith)
                  caseof mwCurrent:
                        go Exit_MoveToNextEntry
                  caseof mwEntire:
                        ddeExecute( ddeCurrent; "[ScrollList(LineDown)]" )
                  caseof mwMarked:
                        ddeExecute( ddeCurrent; "[ScrollList(MarkDown)]" )
            endswitch

            //check for end of list
            currentEntry = strnum( ddeRequest( ddeCurrent; "GetListEntryIndex"
) )
            if (currentEntry = lastEntry)
                  go Exit_MoveToNextEntry
            else
                  lastEntry = currentEntry
            endif

            //test for valid client/contact
            clientID = ddeRequest( ddeCurrent; "Identification")
            entryValid = (clientID <> "")
      endwhile
      retValue = true

      Label( Exit_MoveToNextEntry )
      return( retValue )
endfunc


//*****************************************************************
//logs the note to Maximizer
//*****************************************************************
function LogNote( note; sendTo )
      retValue = true
      if (eMailExist = 0)
            Go (Exit_LogNote)
      endif
      onError( Exit_LogNote )

      if ( AttemptToLogNote )
            if ( sendTo = stPrinter)
                  ddePoke( ddeCurrent; "InsertMailMergeNote"; note )
            endif
            if ( sendTo = stEMail)
                  ddePoke( ddeCurrent; "InsertEMailNote"; note )
            endif
```

```
            if ( not NoteAlreadyTested )
                  NoteAlreadyTested = true

                  e_msg = ddeRequest( ddeCurrent; "CurrentRecordError" )
                  p = strPos( e_msg; " " )
                  e_num = substr( e_msg; 1; p - 1 )

                  if ( e_num <> "0" )
                        e_msg = substr( e_msg; p + 1; strLen(e_msg) - p)

                        messageBox( vDlg; "Maximizer Template"; e_msg + "
Unable to log the note. Do you wish to continue sending anyway?"; YESNO! |
IconExclamation!)
                        if ( vDlg = 6 )
                              AttemptToLogNote = false
                        else
                              retValue = false
                        endif

                  endif
            endif
      endif

      Label( Exit_LogNote )
      return( retValue )
endfunc


//*****************************************************************
//returns the number of entries in the currently active list
//*****************************************************************
function ListCount()
      retValue = 0
      onError( Exit_ListCount)

      retValue = strnum( ddeRequest( ddeCurrent; "GetListCount" ) )

      Label( Exit_ListCount )
      return( retValue )
endfunc


//*****************************************************************
//returns true if the current entry is marked
//*****************************************************************
function IsMarked()
      retValue = false
      onError( Exit_IsMarked)

      retValue = ( ddeRequest( ddeCurrent; "IsMarked" ) = "TRUE" )

      Label( Exit_IsMarked )
      return( retValue )
endfunc


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

```
//waits until the current doc has finished printing
//*****************************************************************
procedure WaitForPrintProcessToFinish()
      DLLLOAD(hUser; "user.exe")

      if (hUser > 32)
            vClass = "WPWPRINT"
            repeat
                  DllCall(hUser;"FindWindow"; vResult: word;
{ANSIString(vClass); 0})
            until (vResult = 0)
      endif

      DLLFREE(hUser)
endproc


//*****************************************************************
//Make sure that the print spooler exits when it's finished - this
//ensures that the WaitForPrintProcessToFinish process works
//*****************************************************************
procedure SetPrintSpooler()
      BIFREAD(
            Status: vResult;
            Group: "WP Print Process";
            Section: "Main Window Options";
            Item: "Exit When Done";
            Value: AutoExitPrintProcess;
      )

      AutoExitPrintProcess = FALSE
      if (AutoExitPrintProcess <> True)
            SetAutoExit(True)
      endif
endproc


//*****************************************************************
//Restore the "Exit When Done" option of the print spooler to its
//original value
//*****************************************************************
procedure ResetPrintSpooler()
      SetAutoExit(AutoExitPrintProcess)
endproc


//*****************************************************************
//Set the value of the "Exit When Done" option of the print spooler
//*****************************************************************
procedure SetAutoExit( vAutoValue)
      repeat
            BifWrite(
                  Status: vStatus;
                  Group: "WP Print Process";
                  Section: "Main Window Options";
                  Item: "Exit When Done";
                  Value: vAutoValue;
```

```
                  ItemType: Boolean!;
            )

            BIFREAD(
                  Status: vResult;
                  Group: "WP Print Process";
                  Section: "Main Window Options";
                  Item: "Exit When Done";
                  Value: vTestValue;
            )
      until (vTestValue = vAutoValue)
endproc


//*************************************************************
//Set the value of the "Exit When Done" option of the print spooler
//*************************************************************
function SaveFileIfNecessary()
      retValue = false
      OnError( Exit_SaveFileIfNecessary )

      if (?Name = "")
            OnCancel(Skip_DoLink)
            FileSaveAsDlg()
            retValue = true
            Label(Skip_DoLink)
      else
            if (?DocModified) FileSave endif
            retValue = true
      endif

      Label(Exit_SaveFileIfNecessary )
      return( retValue )
endfunc


////////////////////////////////////////////////////////////
//
// Shared Functions

//*********************************************************************
//returns true if Maximizer is loaded, false otherwise
//*********************************************************************
function MaximizerIsLoaded()
      isLoaded = false
      onError( Exit_MaximizerIsLoaded )

      isLoaded = ( AppLocate( "Maximizer*" ) <> 0 )

      if ( not isLoaded )
            messagebox( vReturn; "Maximizer Template"; "Please start
Maximizer."; OK! | IconExclamation! )
      endif

Label(Exit_MaximizerIsLoaded)
      return( isLoaded )
```

```
endfunc


//*****************************************************
// Returns true if Maximizer is busy, false otherwise
//*****************************************************
function MaximizerIsBusy()
      returnValue = true
      OnError( Exit_MaximizerIsBusy )

      ddeCurrent = InitiateMaxWinDDE( "SYSTEM" )
      if (ddeCurrent <> 0)
            onError( MaximizerIsBusyDDETerminate )
            returnValue :=  (DDERequest( ddeCurrent; "BUSYINDICATOR" ) =
"BUSY")

            Label( MaximizerIsBusyDDETerminate )
            ddeTerminate( ddeCurrent )
            ddeCurrent = 0
      endif

Label( Exit_MaximizerIsBusy )
      return( returnValue )
endfunc


//*****************************************************
//loops until Maximizer is ready, or until User presses cancel
//returns false if user cancel, else returns true
//*****************************************************
function LoopUntilMaximizerIsReady()
      vResult = 1
      onError( Exit_LoopUntilMaximizerIsReady )

      while ( ( vResult = 1 ) and ( MaximizerIsBusy() ) )
            messagebox( vResult; "Maximizer Template";"Maximizer is currently
busy with another task. Please complete all tasks within Maximizer and click
OK to continue."; OKCANCEL! | IconExclamation! )
      endwhile

Label( Exit_LoopUntilMaximizerIsReady )
      return( vResult = 1 )
endfunc


//*****************************************************
//starts a DDE conversation with MaxWin
//*****************************************************
function InitiateMaxWinDDE(topic)
      dde = 0
      onError( Error_InitiateMaxWinDDE )

Label( Retry_InitiateMaxWinDDE )
      dde = DDEInitiate( "MAXWIN"; topic )
      if ( dde <> 0) go( Exit_InitiateMaxWinDDE ) endif

Label( Error_InitiateMaxWinDDE )
```

```
      messagebox( vReturn; "Maximizer Template"; "Unable to initiate a
conversation with Maximizer. Please verify that Maximizer is running, that a
Maximizer database is open, and that no dialogs are currently active.";
OKCANCEL! | IconExclamation! )
      if ( vReturn = 1 )
            go( Retry_InitiateMaxWinDDe )
      endif

Label( Exit_InitiateMaxWinDDE )
      return( dde )
endfunc


//*****************************************************
//returns true if the current Maximizer list is empty
//*****************************************************
function ListIsEmpty()
      isEmpty = true
      onError( Exit_ListIsEmpty )

      ddeCurrent = InitiateMaxWinDDE( "CURRENTRECORD" )
      if ( ddeCurrent = 0 ) go Exit_ListIsEmpty endif

      onError( ListIsEmptyDDETerminate )
      dummy = ddeRequest( ddeCurrent; "GetListCount" ) = "0"
      isEmpty = false

Label( ListIsEmptyDDETerminate )
      ddeTerminate( ddeCurrent )
      ddeCurrent = 0
      if ( isEmpty ) messageBox( vResult; "Maximizer Template"; "The Maximizer
list is empty."; OK! | IconExclamation! ) endif

Label( Exit_ListIsEmpty )
      return ( isEmpty )
endfunc


//*****************************************************
//displays help
//*****************************************************
function DisplayHelp(contextID)
      dllcall prototype WinHelpA("User32.dll"; ; integer; {hwnd;
ansistring(helpfile); command; data})

      r = winhelpA( 0;"maxwin.hlp"; 1; contextID)
endfunc

////////////////////////////////////////////////////////////////////
//
//MMW: newly added functions
//*****************************************************
//returns true if the current entry does not have an ID
//*****************************************************
function PersonalHotList()
      retValue = false
```

```
      ddeCurrent = InitiateMaxWinDDE("CurrentRecord")
      if (ddeCurrent <> 0)
            if (ddeRequest( ddeCurrent; "Identification") = "")
                  retValue = true
                  messageBox( vDlg; "Maximizer Template"; "This action cannot
be performed when a personal HotList task is selected."; OK! |
IconExclamation!)
            endif
            ddeTerminate( ddeCurrent )
            ddeCurrent = 0
      endif
      return( retValue )
endfunc


//*****************************************************
//returns true if the current entry does not have an ID
//*****************************************************
function DatabaseIsOpen()
      retValue = false

      ddeSystem = InitiateMaxWinDDE( "SYSTEM" )
      if (ddeSystem <> "")
            dbPath = ddeRequest( ddeSystem; "CurrentDatabasePath")
            if (dbPath <> "n/a")
                  retValue = true
            else
                  messagebox( vDlg; "Maximizer Template"; "Please open a
Maximizer Address Book Folder."; OK! | IconExclamation!)
            endif
      endif
      return( retValue )
endfunc
```