

```

//*****
//*   MACRO: ABBREV.WCM
//*   PURPOSE: Allows user to manipulate multiple abbreviations.
//*****
Application (A1; "WordPerfect"; Default; "US")
Display(On!)
InhibitInput(Off!)

                                     // Set the default template here...
WhichTemplate:=0                     //0=current, 1=default, 2=supplemental
UniqueIdentifier:="REC:"             // Identifies records in address book
AllDone:=False

If(WhichTemplate=0) // Sets default radio button
    CurTempl:=1
Else
    If(WhichTemplate=1)
        DefTempl:=1
    Else
        SupTempl:=2
    EndIf
EndIf

If(?BlockActive>0)
    SelectIsOn:=False
Else
    SelectIsOn:=True
EndIf

DllLoad(UserLink; "User")
OnCancel(End@)
OnError(End@)

Dlg:="AbbreviationDialog"
DialogDefine(Dlg; 50; 50; 173; 144; 16+8+2; "Abbreviations")
DialogAddText(Dlg; "T1"; 8; 8; 50; 10; 0; "&Abbreviations:")
DialogAddListBox(Dlg; "Lb1"; 8; 22; 100; 100; 1+32; ListItem)
DialogHandle(hwndList; Dlg; "Lb1")
Call(FillListBox@)
DialogAddPushButton(Dlg; "Pb1"; 116; 8; 43; 13; 1; "C&reate...")
DialogAddPushButton(Dlg; "Pb2"; 116; 24; 43; 13; 0; "&Expand")
DialogAddPushButton(Dlg; "Pb3"; 116; 40; 43; 13; 0; "Re&place")
DialogAddPushButton(Dlg; "Pb4"; 116; 56; 43; 13; 0; "&Delete")
RegionMoveWindow(Dlg+".CancelBttn"; 116; 72; 43; 13)
RegionSetWindowText(Dlg+".CancelBttn"; "&Close")
DialogAddPushButton(Dlg; "Pb6"; 116; 93; 43; 13; 0; "&Location...")
DialogDisplay(Dlg; "Lb1"; Callback@)

```

```

DialogHandle(hwndPb1;Dlg;"Pb1")
DialogHandle(hwndPb2;Dlg;"Pb2")
DialogHandle(hwndPb3;Dlg;"Pb3")
DialogHandle(hwndPb4;Dlg;"Pb4")

// Get count of items in listbox
DllCall(UserLink;"SendMessage";LBCount:INTEGER;{LoWord(hwndList);LoWord(1036);
LoWord(0);0})
If(LBCount>0)
    ItemSelected:=False
Else
    ItemSelected:=True
EndIf
Button3:=True

//*****
/*    Main Loop
//*****
Repeat
If(SelectIsOn)
    If(?BlockActive=0)
        DllCall(UserLink;"EnableWindow";bVoid:BOOL;{LoWord(hwndPb1);
LoWord(0)})
        SelectIsOn:=False
        Button3:=True
    EndIf
Else
    If(?BlockActive>0)
        DllCall(UserLink;"EnableWindow";bVoid:BOOL;{LoWord(hwndPb1);
LoWord(1)})
        SelectIsOn:=True
        Button3:=True
    EndIf
EndIf

// Get count of items in listbox
DllCall(UserLink;"SendMessage";LBCount:INTEGER;{LoWord(hwndList);LoWord(1036);
LoWord(0);0})
If(ItemSelected)
    If(LBCount=0)
        DllCall(UserLink;"EnableWindow";bVoid:BOOL;{LoWord(hwndPb2);
LoWord(0)})
        DllCall(UserLink;"EnableWindow";bVoid:BOOL;{LoWord(hwndPb4);
LoWord(0)})
        ItemSelected:=False
        Button3:=True

```

```

    EndIf
Else
    If(LBCount>0)
        DllCall(UserLink; "EnableWindow"; bVoid:BOOL; {LoWord(hwndPb2);
        LoWord(1)})
        DllCall(UserLink; "EnableWindow"; bVoid:BOOL; {LoWord(hwndPb4);
        LoWord(1)})
        ItemSelected:=True
        Button3:=True
    EndIf
EndIf
If(Button3)
    If(SelectIsOn And ItemSelected)
        DllCall(UserLink; "EnableWindow"; bVoid:BOOL; {LoWord(hwndPb3);
        LoWord(1)})
    Else
        DllCall(UserLink; "EnableWindow"; bVoid:BOOL; {LoWord(hwndPb3);
        LoWord(0)})
    EndIf
    Button3:=False
EndIf
If(WhichTemplate=0 And CheckCurTemplate<>?CurrentTemplate)
    Call(FillListBox@)
EndIf
Until(AllDone)
/*****
/*    End of Main Loop
*****/

Label(End@)          // End of macro
DialogUndisplay(Dlg; "Pb1")
DialogDestroy(Dlg)
DllFree(UserLink)
Quit

/*****
/*    ROUTINE: Callback@
/*    INPUT VARIABLES: Callback[]
/*    OUTPUT VARIABLES: AllDone
/*    DESCRIPTION: This is the callback routine for the dialog. The ID of the control that
was clicked is in Callback[3]
*****/
Label(Callback@)
Switch(Callback[3])
    CaseOf "":
        If(Callback[5]=274 And Callback[6]=61536)

```

```

        AllDone:=True
    EndIf
    Return
CaseOf "Lb1":
    Return
CaseOf "Pb1": // Create
    Call(CheckTextSelected@)
    If(TextSelected)
        Call(NameDialog@)
        If(DialogResult="1")
            OnError(AlreadyExists@)
            AbbreviationCreate(AbbName; WhichTemplate)
            OnError(End@)
            DialogAddListItem(Dlg; "Lb1"; AbbName)
            // Get count of Items
            DllCall(UserLink; "SendMessage"; LBCount:INTEGER;
                {LoWord(hwndList); LoWord(1036); LoWord(0); 0})
            If(LBCount=1)
                // Select the first item in the list
                DLLCall(Userlink; "SendMessage"; nVoid:INTEGER;
                    {LoWord(hwndList); LoWord(1031); LoWord(0);0})
            EndIf
        EndIf
    EndIf
CaseOf "Pb2": // Expand
    Call(GetAbbName@)
    If(AbbName<>"")
        AbbreviationExpand(AbbName; WhichTemplate)
    EndIf
CaseOf "Pb3": // Replace
    Call(GetAbbName@)
    Call(CheckTextSelected@)
    If(AbbName<>"" And TextSelected)
        MessageBox(MsgBoxResult; "WordPerfect For Windows"; "Replace
            abbreviation with new selection?"; IconQuestion! | YesNo!)
        If(MsgBoxResult=6)
            AbbreviationDelete(AbbName; WhichTemplate)
            AbbreviationCreate(AbbName; WhichTemplate)
        EndIf
    EndIf
CaseOf "Pb4": // Delete
    Call(GetAbbName@)
    If(AbbName<>"")
        // get the number of the current selection
        DLLCall(Userlink; "SendMessage"; nIndex:INTEGER;
            {LoWord(hwndList); LoWord(1033); LoWord(0); 0})
    EndIf

```

```

        // delete it from the listbox
        DLLCall(Userlink; "SendMessage"; nVoid:INTEGER;
        {LoWord(hwndList); LoWord(1027); LoWord(nIndex); 0})
        If(nIndex>0)
            NextSel:=nIndex-1
        Else
            NextSel:=0
        EndIf
        // select the next item
        DLLCall(Userlink; "SendMessage"; nVoid:INTEGER;
        {LoWord(hwndList); LoWord(1031); LoWord(NextSel); 0})
        AbbreviationDelete(AbbName; WhichTemplate)
    EndIf
CaseOf "CancelBttn":      // Close
    AllDone:=True
CaseOf "Pb6":
    OldTemplate:=WhichTemplate
    Call(LocationDialog@)
    If(OldTemplate<>WhichTemplate)
        Call(FillListBox@)
    EndIf
EndSwitch
Return
/*****

/*****
/*  ROUTINE: LocationDialog@
/*  INPUT VARIABLES: CurTempl, DefTempl, SupTempl
/*  OUTPUT VARIABLES: WhichTemplate
/*  DESCRIPTION: Prompts the user to choose a template.
/*****
Label(LocationDialog@)
DialogDefine("LocDlg"; 50; 50; 218; 93; 1+2+16; "Abbreviation Location")
DialogAddGroupBox("LocDlg"; "Gb1"; 8; 4; 197; 50; "&Location")
DialogAddRadioButton("LocDlg"; "Rb1"; 18; 14; 89; 11; "C&urrent Template."; CurTempl)
DialogAddRadioButton("LocDlg"; "Rb2"; 18; 26; 89; 11; "De&fault Template."; DefTempl)
DialogAddRadioButton("LocDlg"; "Rb3"; 18; 37; 89; 11; "&Supplemental Template.";
SupTempl)
DialogAddText("LocDlg"; "T1"; 115; 15; 80; 10; 128; ?CurrentTemplate)
DialogAddText("LocDlg"; "T2"; 115; 27; 80; 10; 128; ?TemplateFile)
DialogAddText("LocDlg"; "T3"; 115; 39; 80; 10; 128; ?TemplateFileSupplemental)
DialogDisplay("LocDlg"; 1)
If(MacroDialogResult=1)
    If(CurTempl=1)
        WhichTemplate:=0
    Else

```

```

        If(DefTempl=1)
            WhichTemplate:=1
        Else
            WhichTemplate:=2
        EndIf
    EndIf
EndIf
DialogDestroy("LocDlg")
Return
/*****

/*****
/*    ROUTINE: FillListBox@
/*    INPUT VARIABLES: hwndList; UserLink; WhichTemplate
/*    OUTPUT VARIABLES: None
/*    DESCRIPTION: Fills a list box with the content of the abbreviations of the specified
template.
/*****
Label(FillListBox@)
WaitMessage:="Retrieving abbreviations"
Call(WaitMessage@)
    // reset content of listbox
DllCall(UserLink; "SendMessage"; ListItem:INTEGER; {LoWord(hwndList); LoWord(1029);
LoWord(0); 0})
GetData(RecordCount; Abbreviation!; Count!; WhichTemplate)
For(RecordNumber; 1; RecordNumber<=RecordCount; RecordNumber+1)
    GetData(RecordName; Abbreviation!; Name!; WhichTemplate; RecordNumber)
    If(SubStr(RecordName; 1; 4)<>UniqueIdentifier)
        DialogAddListItem(Dlg; "Lb1"; RecordName)
    EndIf
EndFor
// Select the first item in the list
DLLCall(Userlink;"SendMessage";nVoid:INTEGER;{LoWord(hwndList);LoWord(1031);
LoWord(0);0})
CheckCurTemplate:=?CurrentTemplate
Call(KillWaitMessage@)
Return
/*****

/*****
/*    ROUTINE: NameDialog@
/*    INPUT VARIABLES: None
/*    OUTPUT VARIABLES: AbbName
/*    DESCRIPTION: Prompts the user for an abbreviation name.
/*****
Label(NameDialog@)

```

```

AbbName:=""
DialogDefine("NameDlg"; 50; 50; 172; 76; 1+2+16; "Create Abbreviation")
DialogAddText("NameDlg"; "T1"; 8; 8; 100; 10; 1; "&Abbreviation Name:")
DialogAddEditBox("NameDlg"; "EditBox1"; 8; 19; 150; 13; 32; AbbName; 31)
DialogDisplay("NameDlg"; "EditBox1")
DialogResult:=MacroDialogResult
DialogDestroy("NameDlg")
Return
//*****

//*****
/*  ROUTINE: GetAbbName@
/*  INPUT VARIABLES: ListItem
/*  OUTPUT VARIABLES: AbbName
/*  DESCRIPTION: Checks whether an abbreviation is selected. If not, a message is
displayed.
//*****
Label(GetAbbName@)
    // Get count of Items
DllCall(UserLink; "SendMessage"; LBCount:INTEGER; {LoWord(hwndList); LoWord(1036);
LoWord(0); 0})
If(LBCount>0)
    // Get the text of the selected item
    DllCall(UserLink; "SendMessage"; ListItem:INTEGER; {LoWord(hwndList);
LoWord(1033); LoWord(0); 0})
    DllCall(UserLink; "SendMessage"; nVoid:INTEGER; {LoWord(hwndList);
LoWord(1034); LoWord(ListItem); Address(WPString(AbbName))})
Else
    AbbName:=""
EndIf
If(AbbName="")
    MessageBox(;"Error - No Abbreviation Selected"; "You must select an abbreviation to
use this option."; IconExclamation!)
EndIf
Return
//*****

//*****
/*  ROUTINE: CheckTextSelected@
/*  INPUT VARIABLES: none
/*  OUTPUT VARIABLES: TextSelected
/*  DESCRIPTION: Checks whether selection is active. If not, a message is displayed.
//*****
Label(CheckTextSelected@)
If(?BlockActive=0)
    MessageBox(;"WordPerfect For Windows"; "Select text in the document before creating

```

```

        an Abbreviation."; IconExclamation!)
    TextSelected:=False
Else
    TextSelected:=True
EndIf
Return
//*****

//*****
/*    ROUTINE NAME: AlreadyExists@
/*    INPUT VARIABLES: None
/*    OUTPUT VARIABLES: None
/*    DESCRIPTION: Allows WordPerfect to display an error message.
//*****
Label(AlreadyExists@)
OnError(End@)
Return
//*****

//*****
/*    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:="WAIT PROMPT"
WaitMessageWidth:=80
DialogDefine(WaitTitle; 50; 50; WaitMessageWidth+22; 43; 16+256; WaitTitle)
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@)
Return
//*****

//*****
/*    ROUTINE: KillWaitMessage
/*    INPUT VARIABLES: none
/*    OUTPUT VARIABLES: none
/*    DESCRIPTION: Turns off wait message.
//*****
Label(KillWaitMessage@)

```



DialogDestroy(WaitTitle)

Return

//\*\*\*\*\*