

```

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

// Set the default template here...
Global WhichTemplate := 0 //0 = current, 4 = default, 2 = supplemental
Global AllDone := False
Global CurTempl; DefTempl; SupTempl; AbbName; hwndList; CheckCurTemplate
Global HadError := False

If (WhichTemplate = 0) // Sets default radio button
    CurTempl := 1
Else
    If (WhichTemplate = 4)
        DefTempl := 1
    Else
        SupTempl := 2
    EndIf
EndIf

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

Global UserLink
UserLink := DllLoad ("User32")

Global Dlg := "AbbrevDlg"
DialogDefine (Dlg; 50; 50; 174; 126; 16 + 8; "QuickWords")
DialogAddText (Dlg; "T1"; 8; 8; 50; 10; 0; "&QuickWords:")
DialogAddListBox (Dlg; "Lb1"; 8; 19; 100; 100; 1 + 32; ListItem)
DialogAddPushButton (Dlg; "Pb1"; 116; 8; 50; 14; 1; "C&reate...")
DialogAddPushButton (Dlg; "Pb2"; 116; 26; 50; 14; 0; "&Expand")
DialogAddPushButton (Dlg; "Pb3"; 116; 44; 50; 14; 0; "Re&place")
DialogAddPushButton (Dlg; "Pb4"; 116; 62; 50; 14; 0; "&Delete")
DialogAddPushButton (Dlg; "Pb6"; 116; 80; 50; 14; 0; "&Location...")
DialogAddPushButton (Dlg; "CancelBttn"; 116; 98; 50; 14; CancelBttn!; "&Close")
DialogLoad (Dlg)
hwndList := DialogHandle (Dlg; "Lb1")
WaitMsgInit ()

```

```

If (?CurrentTemplate = ?TemplateFile)
    WhichTemplate := 4 // set current template to default template
EndIf
FillListBox ()
DialogShow (Dlg; "Lb1"; Callback)

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

    // Get count of items in listbox
//DllCall (UserLink; "SendMessageA"; LBCount:INTEGER; {hwndList; 018Bx; 0; 0})
LBCount := RegionGetListCount( hwndList )
If (LBCount>0)
    ItemSelected := False
Else
    ItemSelected := True
EndIf
Button3 := True

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

    // Get count of items in listbox
//DllCall (UserLink; "SendMessageA"; LBCount:INTEGER; {hwndList; 018Bx; 0; 0})
LBCount := RegionGetListCount( hwndList )
If (ItemSelected)
    If (LBCount = 0)
        //DllCall (UserLink; "EnableWindow"; bVoid:BOOL; {hwndPb2; 0})

```

```

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

// End of macro
DialogDismiss (Dlg; "Pb1")
DialogDestroy (Dlg)
DllFree (UserLink)
WaitMsgDestroy ()
Quit

/*****
/*      PROCEDURE: 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]

```
*****
```

```
Procedure Callback ()
```

```
Switch (Callback[3])
```

```
  CaseOf "":
```

```
    If (Callback[5] = 274 And Callback[6] = 61536)
```

```
      AllDone := True
```

```
    EndIf
```

```
    Return
```

```
  CaseOf "Lb1":
```

```
    Return
```

```
  CaseOf "Pb1":// Create
```

```
    If (CheckTextSelected ())
```

```
      If (NameDialog () = "1")
```

```
        OnError Call (AlreadyExists)
```

```
        HadError := False
```

```
        AbbreviationCreate (AbbName; WhichTemplate)
```

```
        OnError ()
```

```
        If (HadError)
```

```
          Return
```

```
        EndIf
```

```
        RegionAddListItem (Dlg + ".Lb1"; AbbName)
```

```
          // Get count of Items
```

```
        //DllCall (UserLink; "SendMessageA"; LBCount:INTEGER;
```

```
        {hwndList; 018Bx; 0; 0})
```

```
        LBCount := RegionGetListCount( hwndList )
```

```
        If (LBCount = 1)
```

```
          // Select the first item in the list
```

```
          //DLLCall (Userlink; "SendMessageA"; nVoid:INTEGER;
```

```
          {hwndList; 0186x; 0; 0})
```

```
          RegionSelectListItemByIndex( Dlg + ".Lb1"; 1; 0 )
```

```
          // index starts with 1, not 0 ???
```

```
        EndIf
```

```
      EndIf
```

```
    EndIf
```

```
  CaseOf "Pb2":// Expand
```

```
    GetAbbName ()
```

```
    If (AbbName <> "")
```

```
      AbbreviationExpand (AbbName; WhichTemplate)
```

```
      //AbbreviationExpand (AbbName; QuickWords!)
```

```
    EndIf
```

```
  CaseOf "Pb3": // Replace
```

```
    GetAbbName ()
```

```
    If (AbbName <> "" And CheckTextSelected ())
```

```
      MessageBox (MsgBoxResult; "WordPerfect For Windows"; "Replace  
QuickWord with new selection?"; IconQuestion! | YesNo!)
```

```

        If (MsgBoxResult = 6)
            AbbreviationDelete (AbbName; WhichTemplate)
            AbbreviationCreate (AbbName; WhichTemplate)
        EndIf
    EndIf
CaseOf "Pb4":        // Delete
    MsgBox (MsgBoxResult; "WordPerfect For Windows"; "Delete QuickWord
from template?"; IconQuestion! | YesNo!)
    If (MsgBoxResult = 6)
        GetAbbName ()
        If (AbbName <> "")
            // get the number of the current selection
            //DLLCall (Userlink; "SendMessageA"; nIndex:INTEGER;
            {hwndList; 0188x; 0; 0})
            nIndex := RegionGetListItem( Dlg + ".Lb1"; AbbName )
            // delete it from the listbox
            RegionRemoveListItem (Dlg + ".Lb1"; AbbName)
            If (nIndex>1) // index starts with 1, not 0 ???
                NextSel := nIndex - 1
            Else
                NextSel := 1
            EndIf
            // select the next item
            //DLLCall (Userlink; "SendMessageA"; nVoid:INTEGER;
            {hwndList; 0186x; NextSel; 0})
            RegionSelectListItemByIndex( Dlg + ".Lb1"; NextSel; 0 )
            AbbreviationDelete (AbbName; WhichTemplate)
        EndIf
    EndIf
CaseOf "CancelBttn": // Close
    AllDone := True
CaseOf "Pb6":
    OldTemplate := WhichTemplate
    LocationDialog ()
    If (OldTemplate <> WhichTemplate)
        FillListBox ()
    EndIf
EndSwitch
Return
EndProc
/*****
/*****
/**  PROCEDURE: LocationDialog
/**  INPUT VARIABLES: CurTempl, DefTempl, SupTempl
/**  OUTPUT VARIABLES: WhichTemplate

```

```

/**      DESCRIPTION: Prompts the user to choose a template.
/*******
Procedure LocationDialog ()
DialogDefine ("LocDlg"; 50; 50; 213; 79; 1 + 2 + 16; "QuickWord 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)
If (?TemplateFileSupplemental = "")
    DialogLoad ("LocDlg"; Dlg)
    RegionEnableWindow ("LocDlg.Rb3"; Disable!)
EndIf
DialogShow ("LocDlg"; Dlg)
If (MacroDialogResult = 1)
    If (CurTempl = 1)
        WhichTemplate := 0
    Else
        If (DefTempl = 1)
            WhichTemplate := 4
        Else
            WhichTemplate := 2
        EndIf
    EndIf
EndIf
If (?CurrentTemplate = ?TemplateFile)
    WhichTemplate := 4 // set current template to default template
EndIf
DialogDestroy ("LocDlg")
Return
EndProc
/*******

/*******
/**      PROCEDURE: FillListBox
/**      INPUT VARIABLES: hwndList; UserLink; WhichTemplate
/**      OUTPUT VARIABLES: None
/**      DESCRIPTION: Fills a list box with the content of the abbreviations (QuickWords) of
the specified template.
/*******
Procedure FillListBox ()
WaitMsgDisplay ("Retrieving QuickWords")
    // reset content of listbox

```

```

RegionResetList (Dlg + ".Lb1")
GetData (RecordCount; Abbreviation!; Count!; WhichTemplate)
For (RecordNumber; 1; RecordNumber <= RecordCount; RecordNumber + 1)
    GetData (RecordName; Abbreviation!; Name!; WhichTemplate; RecordNumber)
    RegionAddListItem (Dlg + ".Lb1"; RecordName)
EndFor
// Select the first item in the list
//DLLCall (Userlink; "SendMessageA"; nVoid:INTEGER; {hwndList; 0186x; 0; 0})
RegionSelectListItemByIndex( Dlg + ".Lb1"; 1; 0 ) // index starts with 1, not 0 ???
CheckCurTemplate := ?CurrentTemplate
WaitMsgHide ()
Return
EndProc
//*****

//*****
/*    FUNCTION: NameDialog
/*    INPUT VARIABLES: None
/*    OUTPUT VARIABLES: AbbName
/*    DESCRIPTION: Prompts the user for an abbreviation (QuickWords) name and returns
the dialog result.
//*****
Function NameDialog ()
AbbName := ""
DialogDefine ("NameDlg"; 50; 50; 166; 58; 1 + 2 + 16; "Create QuickWord")
DialogAddText ("NameDlg"; "T1"; 8; 8; 100; 10; 1; "&QuickWord name:")
DialogAddEditBox ("NameDlg"; "EditBox1"; 8; 19; 150; 13; 32; AbbName; 31)
DialogShow ("NameDlg"; Dlg)
DialogResult := MacroDialogResult
DialogDestroy ("NameDlg")
Return (DialogResult)
EndFunc
//*****

//*****
/*    PROCEDURE: GetAbbName
/*    INPUT VARIABLES: ListItem
/*    OUTPUT VARIABLES: AbbName
/*    DESCRIPTION: Checks whether an abbreviation (QuickWords) is selected. If not, a
message is displayed.
//*****
Procedure GetAbbName ()
    // Get count of Items
//DllCall (UserLink; "SendMessageA"; LBCount:INTEGER; {hwndList; 018Bx; 0; 0})
LBCount := RegionGetListCount( hwndList )
If (LBCount>0)

```

```

        // Get the text of the selected item
        //DllCall (UserLink; "SendMessageA"; ListItem:INTEGER; {hwndList; 0188x; 0; 0})
        //DllCall (UserLink; "SendMessageA"; nVoid:INTEGER; {hwndList; 0189x; ListItem;
        Address (WPString (AbbName))})
        AbbName := RegionGetSelectedText( hwndList )
Else
    AbbName := ""
EndIf
If (AbbName = "")
    MessageBox (; "Error - No QuickWord Selected"; "You must select an QuickWord to use
    this option."; IconExclamation!)
EndIf
Return
EndProc
/*****

/*****
/*     FUNCTION: CheckTextSelected
/*     INPUT: none
/*     OUTPUT: TextSelected
/*     DESCRIPTION: Checks whether selection is active. If not, a message is displayed.
/*****
Function CheckTextSelected ()
If (?BlockActive = 0)
    MessageBox (; "WordPerfect For Windows"; "Select text in the document before creating
    an QuickWord."; IconExclamation!)
    TextSelected := False
Else
    TextSelected := True
EndIf
Return (TextSelected)
EndFunc
/*****

/*****
/*     PROCEDURE NAME: AlreadyExists
/*     DESCRIPTION: Allows WordPerfect to display an error message.
/*****
Procedure AlreadyExists ()
HadError := True
Return
EndProc
/*****

/*****
//     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"; Dlg)
Return
ENDPROC

//*****
//      PROCEDURE: WaitMsgDisplay (Text)
//      PURPOSE: Displays a previously defined wait message dialog.
//*****
PROCEDURE WaitMsgDisplay (Text)
RegionSetWindowText ("WaitMsg" + ".WaitText"; Text)
DialogShow ("WaitMsg"; Dlg; 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
//*****

```