

```

//*****
//      MACRO: ADRSBOOK.WCM
//      PURPOSE: Allows manipulation of the address book.
//*****
Application (A1; "WordPerfect"; Default; "UK")
Display(On!)
DelimiterChar:="|"           // Set general macro variables
FirstTime:=False           // Tests if running macro for first time
NameSelectDlgCreated:=False // State of name selection dialog
InfoEditDlgCreated:=False  // State of info edit dialog
WaitMsgDefined:=False      // State of "please wait message"
Declare(InfoTypeStr[2])
InfoTypeStr[1]:="Address Book"
InfoTypeStr[2]:="Personal Information"
GroupName:="WordPerfect"
SectionName:="Template Information"
ItemName:="Personal Information"
//*****
//      PROGRAM BEGIN
//*****
If(Exists(AdrsbookRunMode)=0)
    AdrsbookRunMode:=0
    Call(WaitMsgInit@)
    Call(WaitMsgDisplay@)
    SelectButton:="&View"
Else
    SelectButton:="&Select"
EndIf
DllLoad(UserLink; "User.exe")
OnError(Error@)
OnCancel(Stop@)
If(Exists(MaxFld[1])=0)
    Global(MaxFld[2])
    MaxFld[1]:=12
    MaxFld[2]:=9
    Global(FldName[2; MaxFld[1]])
    Global(FldVar[2; MaxFld[1]])
    Global(SelectedName[2])
    Global(PreviousSelectedName[2])

    FldName[1; 1]:="Name"           // Set names of address book fields
    FldName[1; 2]:="Salutation"
    FldName[1; 3]:="Title"
    FldName[1; 4]:="Company"
    FldName[1; 5]:="Address"
    FldName[1; 6]:="Town"
    FldName[1; 7]:="County/State"
    FldName[1; 8]:="Postcode"
    FldName[1; 9]:="Country"
    FldName[1; 10]:="Telephone"

```

```

FldName[1; 11]:="Fax"
FldName[1; 12]:="Account/ID"

FldName[2; 1]:="Name" // Set names of personal information
fields
FldName[2; 2]:="Title"
FldName[2; 3]:="Company"
FldName[2; 4]:="Address"
FldName[2; 5]:="Town"
FldName[2; 6]:="County/State"
FldName[2; 7]:="Postcode"
FldName[2; 8]:="Telephone"
FldName[2; 9]:="Fax"
EndIf
Switch(AdrsbookRunMode)
  CaseOf 0; 1:
    InfoType:=1
    UniqueIdentifier:="REC:"
  CaseOf 2; 3:
    InfoType:=2
    UniqueIdentifier:=""
  Default:
    Go(Stop@)
EndSwitch

Switch(AdrsbookRunMode)
  CaseOf 0: // Running Standalone
    Call(NameSelectDlg@)
  CaseOf 1: // Called by macro for address book information
    Call(NameSelectDlg@)
  CaseOf 2: // Called by macro for personal information handling
    If(Exists(SelectedName[InfoType])=0)
      Call(GetPersonal@) // Get personal info
    EndIf
    If(SelectedName[InfoType]="")
      RecordExists:=False
    Else
      RecordExists:=True
    EndIf
    Call(EditInformation@)
  CaseOf 3: // Called by macro for personal information retrieval
    Call(GetPersonal@) // Get personal info
EndSwitch

Label(Stop@)
Call(DestroySelectDlg@)
Call(DestroyInfoEditDlg@)
Call(WaitMsgHide@)
Call(WaitMsgDestroy@)
DllFree(UserLink)

```

```

If(AdrsbookRunMode=0)
    Quit
Else
    Display(Off!)
    Return
EndIf
//*****
//    PROGRAM END
//*****

//*****
//    ROUTINE NAME: GetPersonal@
//    INPUT VARIABLES:
//    OUTPUT VARIABLES:
//    DESCRIPTION: Puts personal information into variables. Prompts for personal
information if none is found.
//*****
Label(GetPersonal@)
If(Exists(SelectedName[InfoType])=0)
    SelectedName[InfoType]:=InfoTypeStr[InfoType]
    Call(RetrieveInformation@)
EndIf
If(Not RecordExists)
    MacroStatusPrompt(On!;"Personalise Your Templates")
    MessageBox("Personalise Your Templates"; "The autofill macro allows you to enter
information about yourself that will personalise your templates. You only need to
enter this information once."; IconInformation!)
    Label(RemoveBlankFields@)
    FirstTime:=True
    Call(EditInformation@)
    FirstTime:=False
    If(InfoEditDlgResult=1)
        For(Count;1;Count<=MaxFld[InfoType];Count+1)
            If(FldVar[InfoType; Count]="")
                MessageBox(MsgBoxResult; "Confirmation"; "One or more
fields are blank. Is this correct?"; IconQuestion! | YesNo!)
                If(MsgBoxResult=7) //NO
                    Go(RemoveBlankFields@)
                Else
                    Break
                EndIf
            EndIf
        EndFor
        Call(MakeSingleString@)
        Call(WriteInformation@)
    Else
        SelectedName[InfoType]:="" // signal that a default was not set.
    EndIf
Else
    If(AdrsBookRunMode=3)

```

```

        Call(RetrieveInformation@)
    EndIf
EndIf
Return
/*****

/*****
//      ROUTINE NAME: NameSelectDlg@
//      INPUT VARIABLES:  You can set SelectedName[] with a default value.
//      OUTPUT VARIABLES: NameSelectDlgResult; SelectedName[]
//      DESCRIPTION:  This routine will present a dialog to the user for the selection of a
//      name from the database.
/*****
Label(NameSelectDlg@)
If(Exists(SelectedName[InfoType]<>0)
    If((SelectedName[InfoType]="")
                                                And
        (Exists(PreviousSelectedName[InfoType]<>0))
        SelectedName[InfoType]:=PreviousSelectedName[InfoType]
    EndIf
EndIf
If(Not NameSelectDlgCreated)
    Call(CreateSelectDlg@)
EndIf
MacroStatusPrompt(On!; "Select or create a record")
DialogDisplay("SelectionDlg"; "List")
NameSelectDlgResult:=MacroDialogResult
Switch(NameSelectDlgResult)
    CaseOf "Be":
                                                // Edit
        If(SelectedName[InfoType]="")
            Go(SelectionErrorPrompt@)
        EndIf
        Call(EditInformation@)
        Go(NameSelectDlg@)
    CaseOf "Bn":
                                                // Add
        Call(NewNameDlg@)
        Go(NameSelectDlg@)
    CaseOf "Bd":
                                                // Delete
        If(SelectedName[InfoType]="")
            Go(SelectionErrorPrompt@)
        EndIf
        Call(DeleteRecord@)
        Go(NameSelectDlg@)
    CaseOf "Bs":
                                                // Select
        If(SelectedName[InfoType]="")
            Go(SelectionErrorPrompt@)
        Else
            Call(RetrieveInformation@)
            If(AdrsbookRunMode <>0)
                PreviousSelectedName[InfoType]:=SelectedName[InfoType]
            Return

```

```

        Else
            Call(NameViewDlg@)
            Go(NameSelectDlg@)
        EndIf
    EndIf
Default:                                     // Close
    MacroStatusPrompt(On!; "Please wait...")
    SelectedName[InfoType]:=""
    Call(ClearFields@)
    Return
EndSwitch
// Execution does not continue here since all cases above either return or jump back.
Label(SelectionErrorPrompt@)
MessageBox(; "Error - No Record Selected"; "You must select a record to use this option.";
IconExclamation!)
Go(NameSelectDlg@)
//*****

//*****
//    ROUTINE NAME: DestroySelectDlg@
//    INPUT VARIABLES: NameSelectDlgCreated
//    OUTPUT VARIABLES: NameSelectDlgCreated
//    DESCRIPTION: Destroys the name selection dialog.
//*****
Label(DestroySelectDlg@)
If(NameSelectDlgCreated)
    DialogDestroy("SelectionDlg")
    NameSelectDlgCreated:=False
EndIf
Return
//*****

//*****
//    ROUTINE NAME: CreateSelectDlg@
//    INPUT VARIABLES: None
//    OUTPUT VARIABLES: NameSelectDlgCreated; hwndList; SelectButton
//    DESCRIPTION: Creates the name selection dialog.
//*****
Label(CreateSelectDlg@)
NameSelectDlgCreated:=True
MacroStatusPrompt(On!; "Please wait...retrieving "+InfoTypeStr[InfoType])
DialogDefine("SelectionDlg"; 50; 50; 198; 161; 16; InfoTypeStr[InfoType])
DialogAddListBox("SelectionDlg"; "List"; 8; 11; 130; 97; 3; SelectedName[InfoType])
GetData(RecordCount; Abbreviation!; Count!; PersonalLibrary!)
For(RecordNumber; 1; RecordNumber<=RecordCount; RecordNumber+1)
    GetData(RecordName; Abbreviation!; Name!; PersonalLibrary!; RecordNumber)
    SubStr(DelPos; 1; 4; RecordName)
    If(DelPos=UniqueIdentifier)
        RecordNameLength:=StrLen(RecordName)
        SubStr(RecordName; 5; RecordNameLength; RecordName)
    
```

```

        DialogAddListItem("SelectionDlg";"List";RecordName)
    EndIf
EndFor
DialogAddPushButton("SelectionDlg"; "Bs"; 146; 9; 38; 13; 1; SelectButton)
DialogAddPushButton("SelectionDlg"; "Be"; 146; 25; 38; 13; 0; "&Edit")
DialogAddPushButton("SelectionDlg"; "Bn"; 146; 41; 38; 13; 0; "&Add")
DialogAddPushButton("SelectionDlg"; "Bd"; 146; 57; 38; 13; 0; "&Delete")
DialogAddPushButton("SelectionDlg"; "Bc"; 146; 73; 38; 13; 0; "&Close")
DialogAddHLine("SelectionDlg"; "H1"; 6; 119; 178)
DialogAddText("SelectionDlg"; "T1"; 8; 121; 176; 20; 0; "Select a name from the list and
press a Function button, or press Add to create a new record.")
DialogHandle(hwndList; "SelectionDlg"; "List")
If(Exists(SelectedName[InfoType])=0)
    // Select the first name in the list if none is already selected
    DLLCall(Userlink; "SendMessage"; nVoid:INTEGER; {LoWord(hwndList);
LoWord(1031); LoWord(0); 0})
EndIf
Return
//*****

//*****
//    ROUTINE NAME: DeleteRecord@
//    INPUT VARIABLES: SelectedName[]
//    OUTPUT VARIABLES: None
//    DESCRIPTION: This routine will delete a record of the database after checking for
confirmation.
//*****
Label(DeleteRecord@)
MessageBox(MsgBoxResult; "Confirmation"; "Delete ""^0""?"; IconQuestion! | YesNo! |
HasParameters!; SelectedName[InfoType])
If(MsgBoxResult=6)
    // get the number of the current selection
    DLLCall(Userlink; "SendMessage"; nIndex:INTEGER; {LoWord(hwndList);
LoWord(1033); LoWord(0); 0})
    // 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 up
    DLLCall(Userlink; "SendMessage"; nVoid:INTEGER; {LoWord(hwndList);
LoWord(1031); LoWord(NextSel); 0})
    AbbreviationDelete(UniqueIdentifier+SelectedName[InfoType]; PersonalLibrary!)
EndIf
Return
//*****

```

```

//*****
//    ROUTINE NAME: RetrieveInformation@
//    INPUT VARIABLES: SelectedName[]
//    OUTPUT VARIABLES: FldVar[]
//    DESCRIPTION: This Routine will read in the information for a selected record if the
record exists.
//*****
Label(RetrieveInformation@)
MacroStatusPrompt(On!; "Please wait...retrieving "+SelectedName[InfoType])
RecordExists:=False
Switch(InfoType)
    CaseOf 0;1:
        GetData(RecordCount; Abbreviation!; Count!; PersonalLibrary!)
        For(RecordNumber;1;RecordNumber<=RecordCount;RecordNumber+1)
            GetData(RecordName; Abbreviation!; Name!; PersonalLibrary!;
RecordNumber)
            If(RecordName=UniqueIdentifier+SelectedName[InfoType])
                RecordExists:=True
                GetData(SingleString; Abbreviation!; Data!; PersonalLibrary!;
RecordNumber)
                Break
            Endif
        EndFor
    CaseOf 2;3:
        BIFRead(BIFStatus; GroupName; SectionName; ItemName; SingleString)
        If(BIFStatus>0)
            RecordExists:=True
        EndIf
EndSwitch
If(RecordExists)
    Call(ParseSingleString@)
Else
    Call(ClearFields@)
EndIf
Return
//*****

//*****
//    ROUTINE NAME: WriteInformation@
//    INPUT VARIABLES: SelectedName[]; InfoType; RecordExists
//    OUTPUT VARIABLES: None
//    DESCRIPTION: Writes a record to the appropriate location..
//*****
Label(WriteInformation@)
Switch(InfoType)
    CaseOf 0;1:
        If(RecordExists) // If the record already existed, then first delete it.
            AbbreviationDelete(UniqueIdentifier+SelectedName[InfoType];
PersonalLibrary!)
        EndIf

```

```

        AbbreviationCreate(UniqueIdentifier+SelectedName[InfoType];
        PersonalLibrary!; SingleString)
    CaseOf 2;3:
        BIFWrite(BIFStatus; GroupName; SectionName; ItemName; SingleString)
EndSwitch
Return
//*****

//*****
//    ROUTINE NAME: EditInformation@
//    INPUT VARIABLES: SelectedName[]
//    OUTPUT VARIABLES: FldVar[]
//    DESCRIPTION: Edits an entry in the address book and prompts user for the input for
//    that entry.
//*****
Label(EditInformation@)
If(InfoType=1)
    Call(RetrieveInformation@)
EndIf
MacroStatusPrompt(On!; "Edit "+InfoTypeStr[InfoType]+" record")
Call(InfoEditDlg@)
If(((InfoType=1) And (InfoEditDlgResult=1))
    Or ((InfoType=2) And (InfoEditDlgResult="PB1")))
    Call(MakeSingleString@)
    Call(WriteInformation@)
Endif
If((InfoType=2) And (InfoEditDlgResult=2))
    Call(RetrieveInformation@)
EndIf
Return
//*****

//*****
//    ROUTINE NAME: NewNameDlg@
//    INPUT VARIABLES: InfoType[];
//    OUTPUT VARIABLES: FldVar[];
//    DESCRIPTION: Creates a new entry in the address book and prompts user for the
//    input for that entry.
//*****
Label(NewNameDlg@)
Call(ClearFields@)
MacroStatusPrompt(On!;"Create "+InfoTypeStr[InfoType]+" record")
Call(InfoEditDlg@)
If(InfoEditDlgResult=2)
    Return
EndIf
Call(MakeSingleString@)
NameToAdd:=FldVar[InfoType; 1]
If(NameToAdd="")
    Go(TryANewName@)

```



```

EndIf
Label(WriteNewName@)
Display(Off!)
OnError(TryANewName@)
AbbreviationCreate(UniqueIdentifier+NameToAdd; PersonalLibrary!; SingleString)
OnError(Error@)
Display(On!)
SelectedName[InfoType]:=NameToAdd
DialogAddListItem("SelectionDlg"; "List"; NameToAdd)
Return

Label(TryANewName@)
OnError(Error@)
DialogDefine("NameDlg"; 50; 50; 172; 94; 19; "Record Name")
DialogAddEditBox("NameDlg"; "EditBox1"; 8; 8; 150; 13; 32; NameToAdd; 27 )
If(NameToAdd="")
    DialogAddText("NameDlg"; "Text1"; 8; 27; 150; 30; 1; "Please enter a unique name
    for this record.")
Else
    DialogAddText("NameDlg"; "Text1"; 8; 27; 150; 30; 1; "A record with this name
    already exists in the "+InfoTypeStr[InfoType]+". Please enter a new name for this
    record.")
EndIf
Display(On!)
DialogDisplay("NameDlg"; "EditBox1")
Display(Off!)
Button:=MacroDialogResult
DialogDestroy("NameDlg")
If(Button<>1) //      check for escape key or cancel button
    Return
Endif
If(NameToAdd="")
    Go(TryANewName@)
Endif
Go(WriteNewName@)
/*****

/*****
//      ROUTINE NAME: InfoEditDlg@
//      INPUT VARIABLES: Any of the string variables may be set to a default value before
calling this function.
//      OUTPUT VARIABLES: FldVar[]; InfoEditDlgResult
//      DESCRIPTION: This Routine will bring up a dialog to edit all of the portions of an
entry into the database.
/*****
Label(InfoEditDlg@)
If(InfoEditDlgCreated)
    Go(DisplayInfoEditDlg@)
EndIf
InfoEditDlgCreated:=True

```

If(InfoType=1)

```
DialogDefine("InfoEditDlg"; 50; 50; 230; 232; 19; InfoTypeStr[InfoType])
DialogAddText("InfoEditDlg"; "T1"; 8; 10; 49; 10; 1; "&Name:")
DialogAddEditBox("InfoEditDlg"; "E1"; 65; 8; 150; 13; 32; FldVar[InfoType; 1]; 80)
DialogAddText("InfoEditDlg"; "T2"; 8; 27; 49; 10; 1; "&Salutation:")
DialogAddEditBox("InfoEditDlg"; "E2"; 65; 25; 150; 13; 32; FldVar[InfoType; 2];
80)
DialogAddText("InfoEditDlg"; "T3"; 8; 44; 49; 10; 1; "&Title:")
DialogAddEditBox("InfoEditDlg"; "E3"; 65; 42; 150; 13; 32; FldVar[InfoType; 3];
80)
DialogAddText("InfoEditDlg"; "T4"; 8; 61; 49; 10; 1; "&Company:")
DialogAddEditBox("InfoEditDlg"; "E4"; 65; 59; 150; 13; 32; FldVar[InfoType; 4];
80)
DialogAddText("InfoEditDlg"; "T5"; 8; 78; 49; 10; 1; "&Address:")
DialogAddEditBox("InfoEditDlg"; "E5"; 65; 76; 150; 13; 32; FldVar[InfoType; 5];
80)
DialogAddText("InfoEditDlg"; "T6"; 8; 95; 49; 10; 1; "To&wn:")
DialogAddEditBox("InfoEditDlg"; "E6"; 65; 93; 150; 13; 32; FldVar[InfoType; 6];
80)
DialogAddText("InfoEditDlg"; "T7"; 8; 112; 53; 10; 1; "C&ounty or State:")
DialogAddEditBox("InfoEditDlg"; "E7"; 65; 110; 150; 13; 32; FldVar[InfoType; 7];
80)
DialogAddText("InfoEditDlg"; "T8"; 8; 129; 49; 10; 1; "&Postcode:")
DialogAddEditBox("InfoEditDlg"; "E8"; 65; 127; 150; 13; 32; FldVar[InfoType; 8];
80)
DialogAddText("InfoEditDlg"; "T9"; 8; 146; 49; 10; 1; "Co&untry:")
DialogAddEditBox("InfoEditDlg"; "E9"; 65; 144; 150; 13; 32; FldVar[InfoType; 9];
80)
DialogAddText("InfoEditDlg"; "T10"; 8; 163; 49; 10; 1; "Tele&phone:")
DialogAddEditBox("InfoEditDlg"; "E10"; 65; 161; 60; 13; 32; FldVar[InfoType; 10];
20)
DialogAddText("InfoEditDlg"; "T11"; 133; 163; 14; 10; 1; "&Fax:")
DialogAddEditBox("InfoEditDlg"; "E11"; 155; 161; 60; 13; 32; FldVar[InfoType; 11];
20)
DialogAddText("InfoEditDlg"; "T12"; 8; 180; 68; 10; 1; "Account/&ID Number:")
DialogAddEditBox("InfoEditDlg"; "E12"; 84; 178; 131; 13; 32; FldVar[InfoType;
12]; 80)
```

Else

```
DialogDefine("InfoEditDlg"; 50; 50; 230; 181; 19; InfoTypeStr[InfoType])
DialogAddText("InfoEditDlg"; "T1"; 8; 10; 49; 10; 1; "&Name:")
DialogAddEditBox("InfoEditDlg"; "E1"; 65; 8; 150; 13; 32; FldVar[InfoType; 1]; 80)
DialogAddText("InfoEditDlg"; "T2"; 8; 27; 49; 10; 1; "&Title:")
DialogAddEditBox("InfoEditDlg"; "E2"; 65; 25; 150; 13; 32; FldVar[InfoType; 2];
80)
DialogAddText("InfoEditDlg"; "T3"; 8; 44; 49; 10; 1; "Company:")
DialogAddEditBox("InfoEditDlg"; "E3"; 65; 42; 150; 13; 32; FldVar[InfoType; 3];
80)
DialogAddText("InfoEditDlg"; "T4"; 8; 61; 49; 10; 1; "&Address:")
DialogAddEditBox("InfoEditDlg"; "E4"; 65; 59; 150; 13; 32; FldVar[InfoType; 4];
80)
```

```

DialogAddText("InfoEditDlg"; "T5"; 8; 78; 49; 10; 1; "&Town:")
DialogAddEditBox("InfoEditDlg"; "E5"; 65; 76; 150; 13; 32; FldVar[InfoType; 4];
80)
DialogAddText("InfoEditDlg"; "T6"; 8; 95; 53; 10; 1; "C&ounty or State:")
DialogAddEditBox("InfoEditDlg"; "E6"; 65; 93; 150; 13; 32; FldVar[InfoType; 4];
80)
DialogAddText("InfoEditDlg"; "T7"; 8; 112; 49; 10; 1; "&Postcode:")
DialogAddEditBox("InfoEditDlg"; "E7"; 65; 110; 150; 13; 32; FldVar[InfoType; 5];
80)
DialogAddText("InfoEditDlg"; "T8"; 8; 129; 49; 10; 1; "Tele&phone:")
DialogAddEditBox("InfoEditDlg"; "E8"; 65; 127; 60; 13; 32; FldVar[InfoType; 6];
20)
DialogAddText("InfoEditDlg"; "T9"; 133; 129; 14; 10; 1; "&Fax:")
DialogAddEditBox("InfoEditDlg"; "E9"; 155; 127; 60; 13; 32; FldVar[InfoType; 7];
20)
If(Not FirstTime)
    DialogAddPushButton("InfoEditDlg";"PB1";8;112;68;13;0;"&Save      as
    Default")
EndIf
EndIf
Label(DisplayInfoEditDlg@)
DialogDisplay("InfoEditDlg"; "E1")
InfoEditDlgResult:=MacroDialogResult
If(InfoEditDlgResult=1)
    Call(CheckForBadEntry@)
    If(BadEntryFound)
        Go(InfoEditDlg@)
    EndIf
EndIf
Return
//*****

//*****
//    ROUTINE NAME: DestroyInfoEditDlg@
//    INPUT VARIABLES: InfoEditDlgCreated
//    OUTPUT VARIABLES: None
//    DESCRIPTION: Destroys the InfoEdit Dlg.
//*****
Label(DestroyInfoEditDlg@)
If(InfoEditDlgCreated)
    DialogDestroy("InfoEditDlg")
    InfoEditDlgCreated:=False
EndIf
Return
//*****

//*****
//    ROUTINE NAME: CheckForBadEntry@
//    INPUT VARIABLES: FldVar[]; MaxFld[]; LookFlds[]
//    OUTPUT VARIABLES: BadEntryFound

```

```
// DESCRIPTION: This routine will check through the strings created in the edit dialog
for DelimiterChar. If it is found, a prompt will ask the user to remove it from the offending
string.
```

```
*****
```

```
Label(CheckForBadEntry@)
```

```
BadEntryFound:=False
```

```
For(Count; 1; Count<=MaxFld[InfoType]; Count+1)
```

```
  StrPos(DelPos;DelimiterChar;FldVar[InfoType; Count])
```

```
  If(DelPos<>0)
```

```
    StringWithBadChar:=FldName[InfoType; Count]
```

```
    BadEntryFound:=True
```

```
    MessageBox( "Illegal Character in "+StringWithBadChar+" Field"; "You
cannot use the pipe character (vertical bar) in any of the information. Please
remove the character from the ^0 field."; IconExclamation! | HasParameters!;
```

```
StringWithBadChar)
```

```
    Return
```

```
  EndIf
```

```
EndFor
```

```
Return
```

```
*****
```

```
*****
```

```
// ROUTINE NAME: MakeSingleString@
```

```
// INPUT VARIABLES: FldVar[]; MaxFld[]
```

```
// OUTPUT VARIABLES: SingleString
```

```
// DESCRIPTION: This function takes all of the separate strings from the dialog and
concatenates them together with separation characters for storage.
```

```
*****
```

```
Label(MakeSingleString@)
```

```
MacroStatusPrompt(On!; "Please wait...writing record to "+InfoTypeStr[InfoType])
```

```
SingleString:=""
```

```
For(Loop; 1; Loop<MaxFld[InfoType]; Loop+1)
```

```
  SingleString:=SingleString+FldVar[InfoType; Loop]+DelimiterChar
```

```
EndFor
```

```
SingleString:=SingleString+FldVar[InfoType; MaxFld[InfoType]]
```

```
Return
```

```
*****
```

```
*****
```

```
// ROUTINE NAME: ParseSingleString@
```

```
// INPUT VARIABLES: SingleString; MaxFld[]
```

```
// OUTPUT VARIABLES: FldVar[]
```

```
// DESCRIPTION: This routine will take the variable SingleString and will parse out
the individual variables for the edit dialog, etc.
```

```
*****
```

```
Label(ParseSingleString@)
```

```
PullString:=SingleString
```

```
For(LoopCounter; 1; LoopCounter<=MaxFld[InfoType]; LoopCounter+1)
```

```
  StrPos(NextDelimLoc; DelimiterChar; PullString)
```

```
  StrLen(PullStringLength; PullString)
```

```

        If(NextDelimLoc=0)
            NextDelimLoc:=PullStringLength+1
        EndIf
        SubStr(FldVar[InfoType; LoopCounter]; 1; NextDelimLoc-1; PullString)
        SubStr(PullString; NextDelimLoc+1; PullStringLength; PullString)
    EndFor
    Return
    /*******

    /*******
    //    ROUTINE: ClearFields
    //    INPUT VARIABLES: FldVar[]; MaxFld[]
    //    OUTPUT VARIABLES: FldVar[]
    //    DESCRIPTION: Sets contents of FldVar[] to all null strings.
    /*******
    Label(ClearFields@)
    For(Loop; 1; Loop<=MaxFld[InfoType]; Loop+1)
        FldVar[InfoType; Loop]:=""
    EndFor
    Return
    /*******

    /*******
    /***    ROUTINE NAME: NameViewDlg
    /***    INPUT VARIABLES:
    /***    OUTPUT VARIABLES:
    /***    DESCRIPTION:
    /*******
    Label(NameViewDlg@)
    Call(CombineInformation@)
    InhibitInput(Off!)    // allow pasting into document
    DialogDefine("LookUpDlg"; 50; 50; 191; 141; 17+8; "Address Book")
    DialogAddGroupBox("LookUpDlg"; "GroupBox1"; 8; 4; 170; 95; SelectedName[InfoType])
    DialogAddEditBox("LookUpDlg"; "Eb1"; 18; 16; 150; 75; 96; CombinedInformation; 1024)
    DialogAddPushButton("LookUpDlg"; "Pb1"; 8; 106; 38; 13; 0; "&Edit")
    DialogAddPushButton("LookUpDlg"; "Pb2"; 54; 106; 68; 13; 0; "&Copy To Clipboard")
    Label(ViewDialogDisplay@)
    Call(WaitMsgHide@)
    DialogDisplay("LookUpDlg";1)
    Call(WaitMsgDisplay@)
    Button:=MacroDialogResult
    Switch(Button)
        CaseOf "Pb1":// Edit
            Call(InfoEditDlg@)
            If(InfoEditDlgResult=1)
                Call(MakeSingleString@)
                Call(WriteInformation@)
            Endif
        CaseOf "Pb2":// Copy to Clipboard
            DialogHandle(hwndEb1; "LookUpDlg"; "Eb1")

```

```

        // Find out what is selected
        DLLCall(UserLink,"SendMessage"; nSelText:INTEGER;
        {LoWord(hwndEb1); LoWord(1024); LoWord(0); 0})
        If(nSelText=0) // If nothing is, then select everything
            DLLCall(UserLink; "SendMessage"; nVoid:INTEGER;
            {LoWord(hwndEb1); LoWord(1025); LoWord(0); 4294901760})
        EndIf
        DLLCall(UserLink; "SendMessage"; nVoid:INTEGER; {LoWord(hwndEb1);
        LoWord(769); LoWord(0); 0})
        Go(ViewDialogDisplay@)
EndSwitch
DialogDestroy("LookUpDlg")
InhibitInput(On!)
Return
//*****

//*****
/*    ROUTINE NAME: CombineInformation
/*    INPUT VARIABLES: FldVar[]
/*    OUTPUT VARIABLES: CombinedInformation
/*    DESCRIPTION: This function takes all of the separate strings from the dialog and
concatenates them together with HardReturn characters for display, etc.
//*****
Label(CombineInformation@)
    // Create a Hrt in a variable
    HRTStr:=NToC(63754)
    CombinedInformation:=""
    If(FldVar[InfoType; 1]<>"")
        CombinedInformation:=CombinedInformation+FldVar[InfoType; 1]+HRTStr
    EndIf
    If(FldVar[InfoType; 3]<>"")
        CombinedInformation:=CombinedInformation+FldVar[InfoType; 3]+HRTStr
    EndIf
    If(FldVar[InfoType; 4]<>"")
        CombinedInformation:=CombinedInformation+FldVar[InfoType; 4]+HRTStr
    EndIf
    If(FldVar[InfoType; 5]<>"")
        CombinedInformation:=CombinedInformation+FldVar[InfoType; 5]+HRTStr
    EndIf
    If(FldVar[InfoType; 6]<>"")
        CombinedInformation:=CombinedInformation+FldVar[InfoType; 6]+HRTStr
    EndIf
    CombinedInformation:=CombinedInformation+HRTStr
    If(FldVar[InfoType; 7]<>"")
        CombinedInformation:=CombinedInformation+FldVar[InfoType; 7]+HRTStr
    EndIf
    If(FldVar[InfoType; 8]<>"")
        CombinedInformation:=CombinedInformation+FldVar[InfoType; 8]+HRTStr
    EndIf
    If(FldVar[InfoType; 9]<>"")

```

```

        CombinedInformation:=CombinedInformation+FldVar[InfoType; 9]
    EndIf
    If(FldVar[InfoType; 10]<>"")
        CombinedInformation:=CombinedInformation+"Telephone: "+FldVar[InfoType; 9]
    EndIf
    If(FldVar[InfoType; 11]<>"")
        CombinedInformation:=CombinedInformation+"Fax: "+FldVar[InfoType; 9]
    EndIf
    If(FldVar[InfoType; 12]<>"")
        CombinedInformation:=CombinedInformation+"ID #: "+FldVar[InfoType; 9]
    EndIf

Return
//*****

//*****
//    ROUTINE NAME: Error@
//    INPUT VARIABLES: None
//    OUTPUT VARIABLES: None
//    DESCRIPTION: Displays a message box when an unknown error occurs.
//*****
Label(Error@)
MessageBox(;"Error"; "An unknown error has occurred. The macro will now terminate.";
IconStop!)
AdrsbookRunMode:=0      // This will cause complete cancellation of macro
Go(Stop@)
//*****

//*****
//    ROUTINE NAME: WaitMsgInit@
//    INPUT VARIABLES: None
//    RETURN VARIABLES: None
//    USES: Nothing
//    DESCRIPTION: Initializes macro wait messages.
//*****
Label(WaitMsgInit@)
WaitDlgId:="WaitAdrsBook"
WaitTitle:="Address Book"
WaitMessage:="Please wait..."
DialogDefine(WaitDlgId;50;50;140;50;16+32;WaitTitle)
DialogAddText(WaitDlgId;"WaitText";8;12;124;10;1+4;WaitMessage)
WaitMsgDefined:=True
Return

Label(WaitDlgCallBack@)
Return
//*****

//*****
//    ROUTINE NAME: WaitMsgDisplay@

```

```

// INPUT VARIABLES: WaitDlgId
// OUTPUT VARIABLES: None
// USES: Nothing
// DESCRIPTION: Displays a previously defined wait message dialog.
//*****
Label(WaitMsgDisplay@)
If(WaitMsgDefined)
    DialogDisplay(WaitDlgId;0;WaitDlgCallBack@)
EndIf
Return
//*****

//*****
// ROUTINE NAME: WaitMsgHide@
// INPUT VARIABLES: WaitDlgId
// OUTPUT VARIABLES: None
// USES: Nothing
// DESCRIPTION: Hides a previously defined wait message dialog.
//*****
Label(WaitMsgHide@)
If(WaitMsgDefined)
    DialogUndisplay(WaitDlgId;1)
EndIf
Return
//*****

//*****
// ROUTINE NAME: WaitMsgDestroy@
// INPUT VARIABLES: WaitDlgId
// OUTPUT VARIABLES: None
// USES: Nothing
// DESCRIPTION: Destroys a previously defined wait message dialog. Hide the dialog
with WaitMsgHide before destroying it.
//*****
Label(WaitMsgDestroy@)
If(WaitMsgDefined)
    DialogDestroy(WaitDlgId)
    WaitMsgDefined:=False
EndIf
Return
//*****

```