

```

//*****
//*   MACRO: ADRS2MRG.WCM
//*   PURPOSE: Copies the address book into a merge data file.
//*****
Application (A1; "WordPerfect"; Default; "US")
UniqueIdentifier:="REC:"
DelimiterChar:="|"
MaxAdrsBookVar:=9
Declare(LookVar[MaxAdrsBookVar])
StopMacro:=False

If(?Substructure)
    MessageBox(; "Not At Main Edit Screen"; "This macro can be played only from the main
    editing screen. Exit all substructures (headers, text boxes, etc.) and play the macro
    again."; IconStop!)
    Go(Quit@)
EndIf
If(Not ?DocBlank)
    MessageBox(MsgBoxResult; "Document Is Not Blank"; "If you continue, the merge data
    will be placed at the current insertion point."; OkCancel! | IconExclamation! |
    DefButton2!)
    If(MsgBoxResult<>1)
        Go(Quit@)
    EndIf
EndIf

WaitMessage:="Inserting address book records into document"
Call(WaitMessage@)
OnError(End@)
OnCancel(End@)

GetData(RecordCount; Abbreviation!; Count!; 1)
For(RecordNumber;1;RecordNumber<=RecordCount;RecordNumber+1)
    GetData(RecordName; Abbreviation!; Name!; 1; RecordNumber)
    If(SubStr(RecordName; 1; 4)=UniqueIdentifier)
        GetData(SingleString; Abbreviation!; Data!; 1; RecordNumber)
        Call(ParseSingleString@)
        MacroStatusPrompt(On!; "Placing "+LookVar[1])
        Call(PlaceFields@)
        If(StopMacro)
            Go(End@)
        EndIf
    EndIf
EndFor

Label(End@)

```

Call(KillWaitMessage@)  
Label(Quit@)  
Quit

```
*****  
/* ROUTINE NAME: PlaceFields@  
/* INPUT VARIABLES: LookVar[]  
/* OUTPUT VARIABLES: None  
/* DESCRIPTION: This routine will place the LookVar[] information into a merge data  
file.
```

```
*****
```

```
Label(PlaceFields@)  
For(FieldNum; 1; FieldNum<=MaxAdrsBookVar; FieldNum+1)  
    Type(LookVar[FieldNum])  
    MergeEndField()  
EndFor  
MergeEndRecord()  
Return
```

```
*****
```

```
*****  
/* ROUTINE NAME: ParseSingleString  
/* INPUT VARIABLES: SingleString  
/* OUTPUT VARIABLES: LookVar[]  
/* 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<=MaxAdrsBookVar; LoopCounter+1)  
    StrPos(NextDelimLoc; DelimiterChar; PullString)  
    StrLen(PullStringLength; PullString)  
    If(NextDelimLoc=0)  
        NextDelimLoc:=PullStringLength+1  
    EndIf  
    SubStr(LookVar[LoopCounter]; 1; NextDelimLoc-1; PullString)  
    SubStr(PullString; NextDelimLoc+1; PullStringLength; PullString)  
EndFor  
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@)
WaitMessageWidth:=149
DialogDefine("Wait"; 50; 50; WaitMessageWidth+22; 72; 16; "Address Book To Merge")
DialogAddText("Wait"; "T1"; 8; 8; 46; 10; 1; "Please Wait...")
DialogAddText("Wait"; "T2"; 8; 19; WaitMessageWidth; 10; 1; WaitMessage)
DialogAddPushButton("Wait"; "Pb1"; (WaitMessageWidth/2)-8; 35; 38; 13; 1; "Cancel")
DialogDisplay("Wait"; "Pb1"; CallBack@)
Return
Label(CallBack@)
If(CallBack[3]="Pb1")
    StopMacro:=True
EndIf
Return
*****
```

```
*****
/* ROUTINE: KillWaitMessage@
/* INPUT VARIABLES: none
/* OUTPUT VARIABLES: none
/* DESCRIPTION: Turns off wait message.
*****
Label(KillWaitMessage@)
DialogUndisplay("Wait"; "Pb1")
DialogDestroy("Wait")
Return
*****
```