1. Use '[Mark] to mark code that I am in the middle of working on and must be changed before final release.
2. The "Review Source Code" feature of MZ-Tools doesn't understand multiple variable declarations on the same line in the (General) section of any module, but it works just fine with multiple variables on the same line within a procedure. So to help that feature be useful, declare all Module and Global scope variables on separate lines.
3. Comments/descriptions of procedures should go INSIDE the procedure block. I tried it above the declaration for a while and didn't like how messy the code looked.
4. All forms should be set to Arial Regular 10pt. The .frm files can be edited in notepad to remove all references to MS Sans Serif (it'll default to the form's font). All labels should be Arial Regular 9pt.
5. All Button\_Click and Menu\_Click events must check that the button is actually Enabled before running the code.
6. Use "On Error Resume Next" in all \_Resize and \_Timer events.
7. All other object events (Form\_Load, Button\_Click, \_MouseMove, etc.) must have error handlers.
8. Never use .SetFocus - use SetFocusWithoutErr <control> instead.
9. ActiveDBInstance may never be accessed from modDB or modCommon.
10. Use this for unexpected events (it is still considered an unhandled error): Err.Raise 1, , "Custom Error".
11. Remember that Unload <form> doesn't unload the code module. Any Private or Static variables will persist to the next time you load the Form, although other stuff like textbox contents will not persist. Creates a dangerous situation. So to be safe, anytime a form in unloaded, also clear the reference so it is removed from memory and must be completely recreated next time:

Unload frmMain

Set frmMain = Nothing

Or, reset it prior to loading it next time:

Set frmMain = New frmMain

frmMain.Show

For safety, in the Form\_Load event of every form/tab in this program, a Private (module-scope) variable will be checked to ensure this is the first time that instance of the form/tab has been loaded.

1. All tab-forms should have the following characteristics:

'Caption = ""

'ControlBox = False

'BorderStyle = None (a 1-pixel border shows in the IDE, but not at runtime)

'KeyPreview = True

'StartupPosition = Manual

'ScaleMode = Pixel

'KeyDown event passed to frmMain

'Do not use \_Load or \_Unload events for anything

1. If you ever jump directly to the CLEAN\_UP label, make sure you increment ERR\_COUNT first:

ERR\_COUNT = ERR\_COUNT + 1: GoTo CLEAN\_UP

1. In the CLEAN\_UP section of every ShowForm variant, Unload Me should be called if there was any error:

If ERR\_COUNT > 0 Then Unload Me

1. The following error handling code templates should be used throughout this program, with the exception of IOLEInPlaceActivate.bas, ITab.cls, and CMNMOD\_ Modules

'errheader>

Const PROC\_NAME = "{MODULE\_NAME}" & "." & "{PROCEDURE\_NAME}": Dim ERR\_COUNT As Integer: On Error GoTo ERR\_HANDLER

'<errheader

{PROCEDURE\_BODY}

CLEAN\_UP:

'Your code here

'errfooter>

Exit {PROCEDURE\_TYPE}

ERR\_HANDLER:

If ERR\_COUNT >= MAXERRS Then: Err.Raise Err.Number, , Err.Description

ERR\_COUNT = ERR\_COUNT + 1: UNHANDLEDERROR PROC\_NAME: Resume CLEAN\_UP

'<errfooter

'errheader>

On Error Resume Next 'ALL ERRORS WILL BE IGNORED IN THIS PROCEDURE

'<errheader

{PROCEDURE\_BODY}

'CUSTOM ERROR HANDLING HERE INSTEAD OF TEMPLATE

''~~\*\*##\*\*~~\*\*##\*\*~~\*\*##\*\*~~\*\*##\*\*~~\*\*##\*\*~~\*\*#

{PROCEDURE\_BODY}

'ANY ERRORS HERE ARE HANDLED BY THE CALLING PROCEDURE

''--..--''--..--''--..--''--..--''--..--''--..--''--.

{PROCEDURE\_BODY}