# Mike’s Test DB Notes

[Forms](#_Forms)

[VBA Scripts](#_VBA_Scripts)

To Do:

* Hide the Access “ribbon” and/or window.
* Add Browser Selection (Radio Buttons & Icons?)

# Forms

|  |  |
| --- | --- |
| **Main** |  |
| **Calendar** |  |

# VBA Scripts

|  |  |
| --- | --- |
| **Forms** | [Form frmCalendar](#_Form_frmCalendar)  [Form frmLogOn](#_Form_frmLogOn)  [Form frmMain](#_Form_frmMain) |
| **Modules** | [Module modGlobal](#_Module_modGlobal)  [Module modUserAccess](#_Module_modUserAccess) |

### Form frmCalendar

'Copyright: Allen Browne's Database And Training, 2007.

'Author: Allen Browne. allen@allenbrowne.com

Option Compare Database

Option Explicit

Private Const conMod = "frmCalendar"

Private Sub cmdCancel\_Click()

On Error GoTo Err\_Handler

'Purpose: Close without transferring date back to calling text box.

DoCmd.Close acForm, Me.Name, acSaveNo

Exit\_Handler:

Exit Sub

Err\_Handler:

Call LogError(Err.Number, Err.Description, conMod & ".cmdCancel\_Click")

Resume Exit\_Handler

End Sub

Private Sub cmdMonthDown\_Click()

Call SetDate("M", -1)

End Sub

Private Sub cmdMonthUp\_Click()

Call SetDate("M", 1)

End Sub

Private Sub cmdOk\_Click()

On Error Resume Next

'Purpose: Transfer the result back to the calling text box (if there is one), and close.

If Me.cmdOk.Enabled Then

If gtxtCalTarget = Me.txtDate Then

'do nothing

Else

gtxtCalTarget = ShowToday(Me.txtDate)

End If

End If

' gtxtCalTarget.SetFocus

DoCmd.Close acForm, Me.Name, acSaveNo

End Sub

Private Sub cmdToday\_Click()

'On Error GoTo Err\_Handler

'Purpose: Set today.

Me.txtDate = Date

Call ShowCal

Exit\_Handler:

Exit Sub

Err\_Handler:

Call LogError(Err.Number, Err.Description, conMod & ".cmdToday\_Click")

Resume Exit\_Handler

End Sub

Private Sub cmdYearDown\_Click()

Call SetDate("YYYY", -1)

End Sub

Private Sub cmdYearUp\_Click()

Call SetDate("YYYY", 1)

End Sub

Private Sub Form\_KeyDown(KeyCode As Integer, Shift As Integer)

'On Error GoTo Err\_Handler

'Purpose:

With Me.txtDate

Select Case KeyCode

Case vbKeyLeft '1 day left or right.

.Value = .Value - 1

KeyCode = 0

Call ShowCal

Case vbKeyRight

.Value = .Value + 1

KeyCode = 0

Call ShowCal

Case vbKeyUp '1 week up or down.

.Value = .Value - 7

KeyCode = 0

Call ShowCal

Case vbKeyDown

.Value = .Value + 7

KeyCode = 0

Call ShowCal

Case vbKeyHome 'Home/End = first/last of this month.

.Value = .Value - Day(.Value) + 1

KeyCode = 0

Call ShowCal

Case vbKeyEnd

.Value = DateSerial(Year(.Value), Month(.Value) + 1, 0)

KeyCode = 0

Call ShowCal

Case vbKeyPageUp 'PgUp/PgDn = previous/next month.

.Value = DateAdd("m", -1, .Value)

KeyCode = 0

Call ShowCal

Case vbKeyPageDown

.Value = DateAdd("m", 1, .Value)

KeyCode = 0

Call ShowCal

Case vbKeyT, vbKeyT + 32 'T or t = today

.Value = Date

KeyCode = 0

Call ShowCal

End Select

End With

Exit\_Handler:

Exit Sub

Err\_Handler:

Call LogError(Err.Number, Err.Description, conMod & ".Form\_KeyDown")

Resume Exit\_Handler

End Sub

Private Sub Form\_Open(Cancel As Integer)

'On Error GoTo Form\_Open\_Err

Dim bEnabled As Boolean

'Initialize to the existing date, or today if null.

' If IsDate(gtxtCalTarget) Then

' Me.txtDate = gtxtCalTarget.Value

' Else

Me.txtDate = Date

' End If

'Lock the Ok button if the text box is locked or disabled.

bEnabled = (gtxtCalTarget.Enabled) And (Not gtxtCalTarget.Locked)

With Me.cmdOk

If .Enabled <> bEnabled Then

.Enabled = bEnabled

End If

End With

'Set the title

If Len(Me.OpenArgs) > 0& Then

Me.Caption = Me.OpenArgs

End If

'Set up the calendar for this month.

Call ShowCal

Form\_Open\_Exit:

Exit Sub

Form\_Open\_Err:

MsgBox Err.Description, vbCritical, "frmCalendar.Form\_Open"

Resume Form\_Open\_Exit

End Sub

Private Function SetSelected(ctlName As String)

On Error GoTo Err\_Handler

Me.txtDate = DateSerial(Year(txtDate), Month(txtDate), CLng(Me(ctlName).Caption))

Call ShowHighligher(ctlName)

Exit\_Handler:

Exit Function

Err\_Handler:

Call LogError(Err.Number, Err.Description, conMod & ".SetSelected")

Resume Exit\_Handler

End Function

Private Function SelectDate(ctlName As String)

Call SetSelected(ctlName)

Call cmdOk\_Click

End Function

Private Function SetDate(Subject As String, Optional intStep As Integer = 1)

On Error GoTo Err\_Handler

Me.txtDate = DateAdd(Subject, intStep, Me.txtDate)

Call ShowCal

Exit\_Handler:

Exit Function

Err\_Handler:

Call LogError(Err.Number, Err.Description, conMod & ".SetDate")

Resume Exit\_Handler

End Function

Private Function ShowCal() As Boolean

On Error GoTo Err\_Handler

'Purpose:

Dim dtStartDate As Date 'First of month

Dim iDays As Integer 'Days in month

Dim iOffset As Integer 'Offset to first label for month.

Dim i As Integer 'Loop controller.

Dim iDay As Integer 'Day under consideration.

Dim bShow As Boolean 'Flag: show label

dtStartDate = Me.txtDate - Day(Me.txtDate) + 1 'First of month

iDays = Day(DateAdd("m", 1, dtStartDate) - 1) 'Days in month.

iOffset = Weekday(dtStartDate, vbSunday) - 2 'Offset to first label for month.

For i = 0 To 41

With Me("lblDay" & Format(i, "00"))

iDay = i - iOffset

bShow = ((iDay > 0) And (iDay <= iDays))

If .Visible <> bShow Then

.Visible = bShow

End If

If (bShow) And (.Caption <> iDay) Then

.Caption = iDay

End If

End With

Next

Call ShowHighligher("lblDay" & Format(Day(Me.txtDate) + iOffset, "00"))

Exit\_Handler:

Exit Function

Err\_Handler:

Call LogError(Err.Number, Err.Description, conMod & ".ShowCal")

Resume Exit\_Handler

End Function

Private Function ShowHighligher(ctlName As String)

On Error GoTo Err\_Handler

Const lngcVOffset As Long = -83

With Me(ctlName)

Me.lblHighlight.Left = .Left

Me.lblHighlight.Top = .Top + lngcVOffset

End With

Exit\_Handler:

Exit Function

Err\_Handler:

Call LogError(Err.Number, Err.Description, conMod & ".ShowHighligher")

Resume Exit\_Handler

End Function

Private Sub lblHelp\_Click()

MsgBox Me.lblHelp.ControlTipText, vbInformation, "Calendar help"

End Sub

### Form frmLogOn

Option Compare Database

Option Explicit

Private Sub cmdEnter\_Click()

On Error GoTo errline

' Check that a User Name is entered

If IsNull(Me.txtUserName) Then

MsgBox "Please enter a User Name.", vbInformation, \_

"Missing User Name"

ResetLogFormTextBoxes

Exit Sub

End If

' Check that a Password is entered

If IsNull(Me.txtPassword) Then

MsgBox "Please enter a Password for the user name " & \_

Me.txtUserName & ".", vbInformation, "Missing User Name"

ResetLogFormTextBoxes

Exit Sub

End If

Dim rs As DAO.Recordset

Set rs = CurrentDb.OpenRecordset("qryRS")

With rs

If .EOF And .BOF Then

MsgBox "Your log on information was incorrect. Try again.", \_

vbCritical, "Failed Log On"

ResetLogFormTextBoxes

GoTo exitline

Else

If !Administrator\_Rights = False Then

DoCmd.OpenForm "frmMain"

DoCmd.Close acForm, "frmLogOn"

GoTo exitline

Else

If cbxEditTable = -1 Then

DoCmd.OpenTable "tblUsers", acViewNormal

DoCmd.Close acForm, "frmLogOn"

Else

DoCmd.OpenForm "frmMain"

DoCmd.Close acForm, "frmLogOn"

End If

End If

End If

End With

exitline:

If Not rs Is Nothing Then

rs.Close

Set rs = Nothing

End If

Exit Sub

errline:

Select Case Err.Number

Case Else

MsgBox "There was an error in the program. Please notify database administrator of the following error: Error Number: " \_

& Err.Number & " " & Err.Description, vbCritical, vbCrLf & "Please write this error down and note what you were doing at the time."

GoTo exitline

End Select

End Sub

Private Sub ResetLogFormTextBoxes()

txtPassword = Null

txtUserName = Null

End Sub

Private Sub txtPassword\_AfterUpdate()

strPW = txtPassword

End Sub

Private Sub txtUserName\_AfterUpdate()

strUser = txtUserName

End Sub

### Form frmMain

Option Compare Database

Private Sub cmdHelp\_Click()

' Open the Database's Description & Help Document!

Dim strFilePath As String

strFilePath = cstrBaseDirectory & cstrHelpFileName

Dim objWord As Word.Application

Set objWord = New Word.Application

With objWord

.Visible = True

.Activate

.WindowState = wdWindowStateNormal

.Documents.Open (strFilePath)

End With

End Sub

Private Sub cmdHyperlinkGo\_Click()

' This uses the default browser. Requires Internet access.

' MsgBox "txtHyperlink is: " & IsNull(txtHyperlink)

If IsNull(txtHyperlink) Or txtHyperlink = "" Then

MsgBox "Please enter a Hyperlink in the text box!"

Else

' MsgBox "txtHyperlink is: " & txtHyperlink

Dim hlk As Hyperlink

Set hlk = [cmdHyperlinkGo].Hyperlink

hlk.Address = Me.txtHyperlink

hlk.Follow

hlk.Address = Null

End If

End Sub

Private Sub cmdQuit\_Click()

' Standard Close the database

DoCmd.Quit

End Sub

Private Sub Form\_Close()

PlaySound ("C:\windows\media\Windows Logoff Sound.wav")

MsgBox "Thanks for using Mike's Test DB!", vbOKOnly, "Mike's Test DB Exit"

End Sub

Private Sub Form\_Load()

If Not (StartUp()) Then

DoCmd.Quit

Else

' Initalize the clock!

Me!lblClock.Caption = Format(Now, "hh:mm:ss AMPM")

' Fill the text boxes in the form.

txtHyperlink = Null

txtUserName = strCurrentUserName

txtComputerName = strMachineName

txtStartDate = strStartUpDate

' Announce the form is open

' MsgBox "PlaySound?", vbOKOnly

If Not PlaySound("C:\windows\media\Windows Logon Sound.wav") Then

MsgBox "Welcome " & strCurrentUserName & " on computer " & strMachineName & ". Today is: " & strStartUpDate

End If

End If

End Sub

Private Sub Form\_Timer()

Me!lblClock.Caption = Format(Now, "hh:mm:ss AMPM")

End Sub

Private Sub lblClock\_DblClick(Cancel As Integer)

' If the clock is static, turn it on, otherwise turn it off!

If Me.TimerInterval = 0 Then

Me.TimerInterval = 1000 ' Timer event every second

' Me.TimerInterval = 1000 \* 60 ' Timer event every minute

' Me.TimerInterval = 1000 \* 360 ' Timer event every hour

Else

Me.TimerInterval = 0

End If

End Sub

Private Sub lblMainTitle\_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)

' MsgBox "Current Color is: " & lblMainTitle.ForeColor

lblMainTitle.ForeColor = &H3C14DC ' RGB(220, 20, 60)

End Sub

Private Sub lblMainTitle\_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)

' MsgBox "Current Color is now: " & lblMainTitle.ForeColor

lblMainTitle.ForeColor = -2147483617

End Sub

### Module modGlobal

Option Compare Database

Option Explicit

' Declare Library Connections

' For GetUserName:

Private Declare Function apiGetUserName Lib "advapi32.dll" Alias \_

"GetUserNameA" (ByVal lpBuffer As String, nSize As Long) As Long

' For GetMachineName:

Private Declare Function apiGetComputerName Lib "kernel32.dll" \_

Alias "GetComputerNameA" \_

(ByVal lpBuffer As String, nSize As Long) As Long

' For PlaySound:

Declare Function apisndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" \_

(ByVal filename As String, ByVal snd\_async As Long) As Long

' Global Constants & Variables

Public strBaseDirectory As String ' Used for directory base

Public strCurrentUserName As String ' Store the Current Users name

Public strStartUpDate As String ' Store the Cureent Startup Date

Public strMachineName As String ' Stores the user's machine name

Private Const cstrMod = "modGlobal"

Public Const cstrBaseDirectory = "C:\Users\mmyers\Documents\03 - Databases\Mike's Test DB\"

Public Const cstrHelpFileName = "Mike's Test DB Notes.docx"

Public Const IsNOFORN = False ' Set to true for NOFORN splash screen notice & acceptance

Public Const IsFOUO = False ' Set to true for FOUO splash screen notice & acceptance

'Calendar form variable:

Public gtxtCalTarget As TextBox 'Text box to return the date from the calendar to.

Public Function CalendarFor(txt As TextBox, Optional strTitle As String)

' On Error GoTo Err\_Handler

' Purpose: Open the calendar form, identifying the text box to return the date to.

' Arguments: txt = the text box to return the date to.

' strTitle = the caption for the calendar form (passed in OpenArgs).

' This is a function so that it can be used for command button expression or macro execution.

Set gtxtCalTarget = txt

DoCmd.OpenForm "frmCalendar", windowmode:=acDialog, OpenArgs:=strTitle

Exit\_Handler:

Exit Function

Err\_Handler:

MsgBox "Error " & Err.Number & " - " & Err.Description, vbExclamation, "CalendarFor()"

Resume Exit\_Handler

End Function

Private Function DisplayFOUOWarning() As Boolean

Dim strSecurityMessage As String

Dim varRetValue As Variant

' Prepare strSecurityMessage for Security Message Display

strSecurityMessage = " \*\*\*FOUO\*\*\*" & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & "This database contains Unclassified Information for Official Use only!" & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & "THIS INFORMATION IS SUBJECT TO SPECIAL EXPORT CONTROLS" & vbCrLf

strSecurityMessage = strSecurityMessage & "AND EACH TRANSMITTAL TO FOREIGN GOVERNMENTS OR FOREIGN" & vbCrLf

strSecurityMessage = strSecurityMessage & "NATIONALS MAY BE MADE ONLY WITH PRIOR APPROVAL OF" & vbCrLf

strSecurityMessage = strSecurityMessage & "COMMANDER NAVAL SEA SYSTEMS COMMAND." & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & "I will comply with all security procedures in the viewing, saving , " & vbCrLf

strSecurityMessage = strSecurityMessage & "sending and printing of this data." & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & " \*\*\*FOUO\*\*\*"

' Display the Security Message and get response

varRetValue = MsgBox(strSecurityMessage, vbOKCancel + vbCritical, \_

"Security Message")

If Not varRetValue = 1 Then

strSecurityMessage = "Unless you agree to abide by security " \_

& "requirements,"

strSecurityMessage = strSecurityMessage & vbCrLf & \_

"you will not be able to access the database!"

varRetValue = MsgBox(strSecurityMessage, vbCritical, \_

"Security Message")

DisplayFOUOWarning = False

Else

DisplayFOUOWarning = True

End If

End Function

Private Function DisplayNOFORNWarning() As Boolean

Dim strSecurityMessage As String

Dim varRetValue As Variant

' Prepare strSecurityMessage for Security Message Display

strSecurityMessage = " \*\*\*NOFORN\*\*\*" & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & "This database contains Sensitive but Unclassified Information" & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & "THIS INFORMATION IS SUBJECT TO SPECIAL EXPORT CONTROLS" & vbCrLf

strSecurityMessage = strSecurityMessage & "AND EACH TRANSMITTAL TO FOREIGN GOVERNMENTS OR FOREIGN" & vbCrLf

strSecurityMessage = strSecurityMessage & "NATIONALS MAY BE MADE ONLY WITH PRIOR APPROVAL OF" & vbCrLf

strSecurityMessage = strSecurityMessage & "COMMANDER NAVAL SEA SYSTEMS COMMAND." & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & "I will comply with all security procedures in the viewing, saving , " & vbCrLf

strSecurityMessage = strSecurityMessage & "sending and printing of this data." & vbCrLf & vbCrLf

strSecurityMessage = strSecurityMessage & " \*\*\*NOFORN\*\*\*"

' Display the Security Message and get response

varRetValue = MsgBox(strSecurityMessage, vbOKCancel + vbCritical, \_

"Security Message")

If Not varRetValue = 1 Then

strSecurityMessage = "Unless you agree to abide by security " \_

& "requirements,"

strSecurityMessage = strSecurityMessage & vbCrLf & \_

"you will not be able to access the database!"

varRetValue = MsgBox(strSecurityMessage, vbCritical, \_

"Security Message")

DisplayNOFORNWarning = False

Else

DisplayNOFORNWarning = True

End If

End Function

' Standard Functions and Subroutines

Public Function GetMachineName() As String

On Error GoTo Err\_Handler

' Purpose: Returns the computername on the network.

' Return: workstation name, or "{Unknown}" on error.

Dim lngLen As Long

Dim lngX As Long

Dim strCompName As String

lngLen = 16&

strCompName = String$(lngLen, 0&)

lngX = apiGetComputerName(strCompName, lngLen)

If lngX <> 0& Then

GetMachineName = Left$(strCompName, lngLen)

Else

GetMachineName = "{unknown}"

End If

Exit\_Handler:

Exit Function

Err\_Handler:

MsgBox Err.Number & ": " & Err.Description & vbCrLf & vbCrLf & \_

"From Procedure: " & cstrMod & ".GetMachineName"

Resume Exit\_Handler

End Function

Public Function GetUserName() As String

' Returns the network login name

Dim lngLen As Long, \_

lngX As Long

Dim strUserName As String

strUserName = String$(254, 0)

lngLen = 255

lngX = apiGetUserName(strUserName, lngLen)

If (lngX > 0) Then

GetUserName = Left$(strUserName, lngLen - 1)

Else

GetUserName = "Name Not Available"

End If

End Function

Public Function LogError(lngErr As Long, strDescrip As String, strProc As String, \_

Optional bShowUser As Boolean = True, Optional varParam As Variant)

'Purpose: Minimal substitute for the real error logger function at:

' http://allenbrowne.com/ser-23a.html

If bShowUser Then

MsgBox "Error " & lngErr & ": " & strDescrip, vbExclamation, strProc

End If

End Function

Public Function PlaySound(sWavFile As String) As Boolean

' Purpose: Plays a sound; Argument: the full path and file name

If apisndPlaySound(sWavFile, 1) = 0 Then

MsgBox "The sound did not play!"

PlaySound = False

Else

PlaySound = True

End If

End Function

Public Function WeekdayName(Optional inDate As Variant) As String

' Returns the name for the day of the week based on any date, if provided

If IsMissing(inDate) Then

inDate = Now()

' MsgBox "No in date provided to WeekdayName, using today's date!"

End If

Dim TodaysDate As String

Select Case Weekday(inDate)

Case vbSunday ' 1

TodaysDate = "Sunday"

Case vbMonday ' 2

TodaysDate = "Monday"

Case vbTuesday ' 3

TodaysDate = "Tuesday"

Case vbWednesday ' 4

TodaysDate = "Wednesday"

Case vbThursday ' 5

TodaysDate = "Thursday"

Case vbFriday ' 6

TodaysDate = "Friday"

Case vbSaturday ' 7

TodaysDate = "Saturday"

End Select

WeekdayName = TodaysDate

End Function

Public Function ShowToday(Optional inDate As Variant) As String

' Creates today's date as a string in YYYY-MM-DD format

If IsMissing(inDate) Then

' MsgBox "No in date provided to ShowToday, using today's date!"

inDate = Now()

End If

Dim TodaysDate As String

TodaysDate = Year(inDate) & "-"

If Month(inDate) < 10 Then

TodaysDate = TodaysDate & "0" & Month(inDate) & "-"

Else

TodaysDate = TodaysDate & Month(inDate) & "-"

End If

If Day(inDate) < 10 Then

TodaysDate = TodaysDate & "0" & Day(inDate)

Else

TodaysDate = TodaysDate & Day(inDate)

End If

ShowToday = TodaysDate

End Function

Function StartUp() As Boolean

' This is the Start Up Function (Can be called by the Autoexec Macro or

' as an OnLoad event for the Main form if desginated for the database.

' Handle security Requirements

StartUp = True

If IsFOUO Then

StartUp = DisplayFOUOWarning

End If

If IsNOFORN Then

StartUp = DisplayNOFORNWarning

End If

If StartUp Then

' Initialize User & Machine global information

strCurrentUserName = GetUserName()

strMachineName = GetMachineName()

strStartUpDate = ShowToday()

' MsgBox "Welcome " & strCurrentUserName & " on computer " & strMachineName & ". Today is: " & strStartUpDate

' Define the Current Database Storage location

strBaseDirectory = "C:\Users\mmyers\Documents\03 - Databases\Mike's Test DB"

RunCommand acCmdWindowHide

' DoCmd.OpenForm "frmMain", acNormal, , , acFormEdit, acWindowNormal ' not needed if called from frmMain OnLoad!

DoCmd.LockNavigationPane True ' Prevent object deletion!

End If

End Function

### Module modUserAccess

Option Compare Database

Option Explicit

' This module is to control user access to the database

Public strUser As Variant

Public strPW As Variant

Public Function GetUser()

GetUser = strUser

End Function

Public Function GetPW()

GetPW = strPW

End Function