```
Option Compare Database

Private Sub Command1_Click()
    DoCmd.OpenForm "frm_MainControl"
    DoCmd.Close acForm, "frm_ActionNotice"
End Sub
```

```
Option Compare Database

Private Sub Command10_Click()
    DoCmd.OpenForm "frm_MainControl"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub


Private Sub Command128_Click()
    DoCmd.OpenForm "frm_ProgramMonthly_Request_BC"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub

Private Sub Command157_Click()
    DoCmd.OpenForm "frm_RequirementOnly_Request_BC"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub

Private Sub Command161_Click()
    DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BC"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub
```

```
Option Compare Database

Private Sub Command10_Click()
    Application.Quit
End Sub


Private Sub Command128_Click()
    DoCmd.OpenForm "frm_ProgramOnly_Request_BO"
    DoCmd.Close acForm, "frm_BO_UnapprovedCount"
End Sub
```

```
Option Compare Database

Private Sub Command10_Click()
    DoCmd.OpenForm "frm_MainControl"
    DoCmd.Close acForm, "frm_BCSub_UnapprovedCount"
End Sub


Private Sub Command128_Click()
    DoCmd.OpenForm "frm_ProgramMonthly_Request_BC_Sub"
    DoCmd.Close acForm, "frm_BCSub_UnapprovedCount"
End Sub

Private Sub Command157_Click()
    DoCmd.OpenForm "frm_RequirementOnly_Request_BC_Sub"
    DoCmd.Close acForm, "frm_BCSub_UnapprovedCount"
End Sub

Private Sub Command161_Click()
    DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BC_Sub"
    DoCmd.Close acForm, "frm_BCSub_UnapprovedCount"
End Sub
```

```
Option Compare Database

Private Sub Command10_Click()
    DoCmd.OpenForm "frm_MainControl"
    DoCmd.Close acForm, "frm_BO_UnapprovedCount"
End Sub


Private Sub Command128_Click()
    DoCmd.OpenForm "frm_ProgramMonthly_Request_BO"
    DoCmd.Close acForm, "frm_BO_UnapprovedCount"
End Sub

Private Sub Command156_Click()
    DoCmd.OpenForm "frm_RequirementOnly_Request_BO"
    DoCmd.Close acForm, "frm_BO_UnapprovedCount"
End Sub

Private Sub Command160_Click()
    DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BO"
    DoCmd.Close acForm, "frm_BO_UnapprovedCount"
End Sub
```

```
Option Compare Database

Private Sub Command10_Click()
    Application.Quit
End Sub


Private Sub Command128_Click()
    DoCmd.OpenForm "frm_ProgramOnly_Request_BO"
    DoCmd.Close acForm, "frm_BO_UnapprovedCount"
End Sub
```

```
Option Compare Database

Private Sub Command0_Click()

    answer = MsgBox("Are you sure you want to close out the fiscal year? This action is not reversable
.", vbYesNo + vbQuestion, "Fiscal Year Close Out")

    If answer = vbYes Then Call closeoutEOY

End Sub

Private Sub Command3_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_MainControl", acNormal
End Sub
```

```
Option Compare Database

Private Sub Command5_Click()
    DoCmd.Close acForm, "frm_CompactandRepair"
    DoCmd.OpenForm "frm_MainControl", acNormal
End Sub

Private Sub CompactandRepairDB_Click()
    If Len(Trim(Dir("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Da
tabase\2_Engines\PRR_Engine\PRR_BE\PRR_BE.laccdb"))) > 0 Then
        MsgBox "Database is locked"
    Else

    If Len(Trim(Dir("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Da
tabase\2_Engines\SOF_Database_BE\SOF_Database_BE.laccdb"))) > 0 Then
        DoCmd.Close acForm, "frm_CompactandRepair"
        DoCmd.OpenForm "frm_MainControl", acNormal
        MsgBox "Database is locked"
        Else

    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\2_Engines\PRR_Engine\PRR_BE\PRR_BE.accdb", "\\RUCKW0U9G67001\drm\PBD SOF Database\7_Backup_E
ngines\PRR_BE_Backup" & (Format(Now, "yyyymmdd_hhmm")) & ".accdb")
    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\2_Engines\SOF_Database_BE\SOF_Database_BE.accdb", "\\RUCKW0U9G67001\drm\PBD SOF Database\7_B
ackup_Engines\SOF_BE_Backup" & (Format(Now, "yyyymmdd_hhmm")) & ".accdb")

    Call updateStatusBar("Running Compact and Repair on SOF_Database_BE.....")
    Path = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\2_
Engines\SOF_Database_BE\"
    DBEngine.CompactDatabase Path & "SOF_Database_BE.accdb", Path & "Temp.accdb", , , ";pwd=g8pae"
    Kill Path & "SOF_Database_BE.accdb"
    Name Path & "Temp.accdb" As Path & "SOF_Database_BE.accdb"

    Call updateStatusBar("Running Compact and Repair on PRR_BE.....")
    Path = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\2_
Engines\PRR_Engine\PRR_BE\"
    DBEngine.CompactDatabase Path & "PRR_BE.accdb", Path & "Temp.accdb"
    Kill Path & "PRR_BE.accdb"
    Name Path & "Temp.accdb" As Path & "PRR_BE.accdb"
    Call clearStatusBar
    DoCmd.Close acForm, "frm_CompactandRepair"
    DoCmd.OpenForm "frm_MainControl", acNormal
    MsgBox "Compact and Repair Complete", vbInformation

End If
End If
End Sub

Private Sub Form_Close()
    DoCmd.OpenForm "frm_MainControl", acNormal
End Sub
```

```
Option Compare Database

Private Sub Cost_Collector_AfterUpdate()
    'Me.Refresh
    'If Nz(Me.[Cost Collector]) = "" Then Exit Sub
    If Len(Me.[Cost Collector]) = 0 Then
        MsgBox "The Cost Collector field blank cannot be blank!"

       'If Not IsNull(SQL = "SELECT [Cost Collector] FROM [Cost Collector Information]" & _
       ' "WHERE [Cost Collector Information].[Cost Collector] = '" & Me.[Cost Collector] & "'") Then
        'MsgBox "You have entered a duplicate value!"
        'Cancel = True
    'End If

    ElseIf Not IsNull(DLookup("[Cost Collector]", "[02_tbl_CostCollectorInformation]", "[Cost Collecto
r] = '" & Me.[Cost Collector] & "'")) Then
        MsgBox "You have entered a duplicate value!" & vbCrLf & "Please type a different value or dele
te this record."
        Me.Undo
        Cancel = True
    End If
End Sub

Private Sub Cost_Collector_BeforeUpdate(Cancel As Integer)
    'Do nothing
End Sub
```

```
Option Compare Database

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & Me.Combo302.Value & " (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)

    End Sub




Private Sub Form_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & Me.Combo302.Value & " (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)

    End Sub



Private Sub Form_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & Me.Combo302.Value & " (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)

    End Sub

Private Sub Form_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Command0_Click()
    DoCmd.OpenForm "frm_MainControl"
    DoCmd.Close acForm, "frm_ExceptionMenu"
End Sub

Private Sub Command3_Click()
    'BO Button

End Sub

Private Sub Command9_Click()
    'BC Button

End Sub
```

```
Option Compare Database




Private Sub Cost_Collector__To__AfterUpdate()
Dim dbs As Database, rst As Recordset
Dim SQL As String
Set dbs = CurrentDb

SQL = "SELECT [Functional Area] FROM [02_tbl_CostCollectorInformation]" & _
        "WHERE [02_tbl_CostCollectorInformation].[Cost Collector] = '" & Me.[Cost Collector (To)] & "'
"

If Me.[Cost Collector (To)] = "" Then
        Me.[Functional Area (To)] = ""
    Else
        Set rst = dbs.OpenRecordset(SQL)
        Me.[Functional Area (To)] = rst![Functional Area]
        '= rst![SQL]
        Me.Refresh
End If

End Sub

Private Sub Form_Close()
DoCmd.SetWarnings (WarningsOff)
    'DoCmd.OpenQuery "qry_Delete_Neg_Fenced"
    DoCmd.OpenQuery "000_Delete_FMT_Neg_Fenced"
    DoCmd.OpenQuery "qry_Append_Neg_Fenced"
    Call exportFMTlog
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.Close , "frm_FunctionalAreas"
    DoCmd.OpenForm "frm_MainControl"
    Call exportFA
End Sub
```

```
Option Compare Database

Private Sub btnEmail_Click()
    'Run function to generate email from modules based on combo box selection
    If (Me!cboTypeBody) = "" Or IsNull(Me!cboTypeBody) Then
        MsgBox "Please choose a type of email to continue.", vbOKOnly, "Not Selected"
    ElseIf (Me!cboTypeBody) = "Daily" Or (Me!cboTypeBody) = "EOM Cleanup" Or (Me!cboTypeBody) = "EOM F
inalized" Then
        Call emailSOF
    ElseIf (Me!cboTypeBody) = "Consumption" Then
        Call emailConsumption
    ElseIf (Me!cboTypeBody) = "Database Update" Then
        Call emailManDBUpdate
    ElseIf (Me!cboTypeBody) = "SAG Recap" Then
        Call emailSAGRecap
    End If
End Sub

Private Sub btnFHPParts_Click()

    Dim cboSelect As String

    If IsNull(Me!cboFHPParts) Then
        MsgBox "No selection was made. Please select an option and try again."
    Else
        cboSelect = Me!cboFHPParts
    End If


    Select Case cboSelect

    'Creates a new file to input the FHP Parts data
    Case "New"

        newFHPParts

    'Opens a file explorer to the appropriate folder with all the completed, but not submitted, FHP Pa
rts files
    Case "Edit/Approve"

        Shell "C:\WINDOWS\explorer.exe """ & "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\
00 PBD SOF Databse\RTS Database\1_Data\FHP - Parts Form" & "", vbNormalFocus

    'If the user is an admin, the user can select a file from the explorer to be submited into the dat
abase
    Case "Submit"

        Dim strSQL As String
        Dim rs As Recordset

        strSQL = ("SELECT * FROM qry_EnvironAdminCheck")
        Set rs = CurrentDb.OpenRecordset(strSQL)
        With rs
            If !Admin = -1 Then

                submitMultiRequestForm ("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PB
D SOF Databse\RTS Database\1_Data\FHP - Parts Form\")

                MsgBox "Operation Complete"

            Else

                MsgBox "You do not have permission to submit these requests. Please contact the databa
se administrator."

            End If

        End With

    'MsgBox that shows how many files have yet to be submitted
    Case "Status"

        excelFormStatus ("FHP - Parts")
    Case Else

        'Do nothing, the null option was already taken care of at the beginning

    End Select

End Sub

Private Sub btnCivPayForm_Click()

    Dim cboSelect As String

    If IsNull(Me!cboCivPayForm) Then
        MsgBox "No selection was made. Please select an option and try again."
    Else
        cboSelect = Me!cboCivPayForm
    End If

    Select Case cboSelect

    'Creates a new file to input the
    Case "New"

        newCivPayForm

    'Opens a file explorer to the appropriate folder with all the completed, but not submitted, FHP Pa
rts files
    Case "Edit/Approve"

        Shell "C:\WINDOWS\explorer.exe """ & "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\
00 PBD SOF Databse\RTS Database\1_Data\CivPay Form" & "", vbNormalFocus

    'If the user is an admin, the user can select a file from the explorer to be submited into the dat
abase
    Case "Submit"

        Dim strSQL As String
        Dim rs As Recordset

        strSQL = ("SELECT * FROM qry_EnvironAdminCheck")
        Set rs = CurrentDb.OpenRecordset(strSQL)
        With rs
            If !Admin = -1 Then

                submitMultiRequestForm ("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PB
D SOF Databse\RTS Database\1_Data\CivPay Form\")

                MsgBox "Operation Complete"

            Else

                MsgBox "You do not have permission to submit these requests. Please contact the databa
se administrator."

            End If

        End With

    'MsgBox that shows how many files have yet to be submitted
    Case "Status"

        excelFormStatus ("CivPay")

    Case Else

        'Do nothing, the null option was already taken care of at the beginning

    End Select

End Sub
```

```
Private Sub BtnPgmReqUpdate_Click()
    DoCmd.OpenForm "frm_Requirements_and_Program", acNormal
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub btnLdrshp_Click()
    Call SOF_DB_Update
    Call SOFBackup

    Call updateStatusBar("SoF Update complete")
    MsgBox "SoF Update complete.", vbOKOnly
    Call clearStatusBar

End Sub

Private Sub btnSpendPlanUpdate_Click()
    'In main form PRR Forms/Reports tab
    Application.FollowHyperlink "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Da
tabse\RTS Database\1_Data\SpendPlan\Spend Plan Update Form.xlsm"
End Sub

Private Sub btnUnitFundingUpdate_Click()
    'In main form PRR Forms/Reports tab

    Dim strSQL As String
    Dim rs As Recordset

    strSQL = ("SELECT * FROM qry_EnvironAdminCheck")
    Set rs = CurrentDb.OpenRecordset(strSQL)
    With rs
        If !BC_Approval = -1 Or !Admin = -1 Then

            Application.FollowHyperlink "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PB
D SOF Databse\RTS Database\1_Data\Unit Funding\Unit Funding Update Form.xlsm"

        Else

            DoCmd.OpenTable "Unit Funding", acViewNormal, acReadOnly

        End If

    End With

End Sub

Private Sub Command104_Click()
  Call SOF_DB_Update
    Call SOFBackup

    Call updateStatusBar("SoF Update complete")
    MsgBox "SoF Update complete.", vbOKOnly
    Call clearStatusBar
'DoCmd.SetWarnings False

'Dim currDB As DAO.Database
'Dim rs As Recordset
'Set currDB = CurrentDb
'Dim SQL As String
'Dim tSQL As String

 '   Call cleanBackups
  '  Call backendPRRBackup
   ' Call backendBackupSOF
    'tSQL = "INSERT INTO tbl_BackupLog (LastBackupTime)" & "VALUES (Now())"
    'DoCmd.RunSQL tSQL
    'SQL = ("SELECT * FROM qry_PRR_Last_BU")
    'Set rs = CurrentDb.OpenRecordset(SQL)
    'With rs
     '   Me.Form.Requery
    'End With
    'Me.Dirty = False
'DoCmd.SetWarnings True
End Sub
```

```
Private Sub Command114_Click()
DoCmd.SetWarnings False
    'Not needed anymore. Saved in case the need comes back.
    'DoCmd.OutputTo acOutputReport, "rpt_OutstandingRequests", acFormatPDF, "\\RUCKW0U9G67001\drm\PBD\
DATA CALL\PAE\Daily Reports\Outstanding Request\Outstanding Requests " & (Format(Now, "mm-dd-yyyy")) &
 ".pdf"   'Save to location
    DoCmd.OpenReport "rpt_OutstandingRequests", acViewReport
    DoCmd.Close acForm, "frm_MainControl"
DoCmd.SetWarnings True
End Sub

Private Sub Command117_Click()
DoCmd.SetWarnings False
    'Updates the Program Details, including the Phased Obligations and Phased Commitments reports.
    'This code runs the queries to delete the old data and import the new data in the appropriate fold
ers
    DoCmd.OpenQuery "000_Delete_PrgmDetails"
    DoCmd.OpenQuery "qry_TIGERProgram_Append"
    DoCmd.OpenQuery "000_Delete_PhaseObReport"
    DoCmd.OpenQuery "qry_PhaseObReport_Append"
    DoCmd.OpenQuery "000_Delete_PhaseCommReport"
    DoCmd.OpenQuery "qry_PhaseCommReport_Append"
    Call exportPrgmDetails
    Call exportPhaseObReport
    Call exportPhaseCommReport
    MsgBox "The Program Details data has been updated.", vbOKOnly, "Update Notice"

DoCmd.SetWarnings True
End Sub

Private Sub Command122_Click()
    DoCmd.OpenForm "frm_TableEdit_ProgramOnly"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command127_Click()
    Call AttachmentPurge
End Sub

Private Sub Command132_Click()
    DoCmd.OpenForm "frm_FunctionalAreas"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command133_Click()
    DoCmd.OpenForm "frm_Requests_Archived"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command139_Click()
    DoCmd.OpenForm "frm_TableEdit_RequirementOnly"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command140_Click()
    DoCmd.OpenForm "frm_TableEdit_PnR"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command156_Click()
    Call TDYupdate
End Sub

Private Sub Command160_Click()
    DoCmd.OpenForm "frm_TDY_CostCollector_128th"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command20_Click()
    DoCmd.OpenForm "frm_Requests_ALL"
    DoCmd.Close acForm, "frm_MainControl"
End Sub
```

```
Private Sub Command25_Click()
    DoCmd.OpenReport "rpt_RequestForm", acViewPreview
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command28_Click()
    DoCmd.OpenForm "frm_ObjectClassUpdate"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command395_Click()
    Call SOF_DB_Update

    Call updateStatusBar("SoF Update complete")
    MsgBox "SoF Update complete.", vbOKOnly
    Call clearStatusBar
End Sub

Private Sub Command400_Click()
    Call SOFBackup

    Call updateStatusBar("SoF Backup complete")
    MsgBox "SoF Backup complete.", vbOKOnly
    Call clearStatusBar
End Sub

Private Sub Command41_Click()
    DoCmd.OpenReport "rpt_Requirements_and_Program", acViewReport
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command480_Click()
    'In main form SOF Interface tab
    Call importSpendPlanFile

End Sub

Private Sub Command48_Click()
    DoCmd.OpenForm "frm_OutlookBody"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command489_Click()
    'In main form SOF Interface tab
    Call importUnitFundingFile

End Sub

Private Sub Command504_Click()

    Call blankRIDQueries

    MsgBox ("Delete successful")

End Sub

Private Sub Command513_Click()

    DoCmd.OpenForm "frm_PgmAndReqFMT_Request", acNormal
    DoCmd.Close acForm, "frm_MainControl"

End Sub

Private Sub Command67_Click()
    If Me.dteFilter = "" Then
        'dateSelect = MsgBox("You must select a date to proceed.", vbOKOnly, "Date Entry Notice")
        DoCmd.OpenReport "rpt_EarningsCompleted", acViewPreview
    Else
        DoCmd.OpenReport "rpt_EarningsCompleted", acViewPreview, , "[Date]=[Forms].[frm_MainControl].[
dteFilter]"
    End If
    Me!dteFilter.Value = ""
```

```
End Sub


Private Sub Command70_Click()
    DoCmd.OpenForm "frm_ParentExtOrg"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub Command75_Click()
    If IsNull(Me.Combo78) Or Me.Combo78 = "" Then
        DoCmd.OpenReport "rpt_TTA_ALLGrp", acViewReport
        DoCmd.Close acForm, "frm_MainControl"
    Else
        DoCmd.OpenReport "rpt_TTA_bySelect", acViewReport
        DoCmd.Close acForm, "frm_MainControl"
    End If
End Sub

Private Sub Command90_Click()
    If Me.Combo86 = "Program" Then
        DoCmd.OpenForm "frm_DocRetrieval_Pgm"
        DoCmd.Close acForm, "frm_MainControl"
    ElseIf Me.Combo86 = "Requirement" Then
        DoCmd.OpenForm "frm_DocRetrieval_Req"
        DoCmd.Close acForm, "frm_MainControl"
    Else
        DoCmd.OpenForm "frm_DocRetrieval_PnR"
        DoCmd.Close acForm, "frm_MainControl"
    End If
End Sub

Private Sub CompactRepairDB_Click()
    DoCmd.Close acForm, "frm_MainControl"
    DoCmd.OpenForm "frm_CompactandRepair", acNormal
End Sub


Private Sub Form_Load()
Dim currDB As DAO.Database
Dim rs As Recordset
Set currDB = CurrentDb
Dim SQL As String

SQL = ("SELECT * FROM qry_EnvironAdminCheck")
    Set rs = CurrentDb.OpenRecordset(SQL)
    With rs
        If !Admin = -1 Then
             Me.Admin.Visible = True
             Me.SOF_Interface.Visible = True
             Me.[PRR Forms/Reports].Visible = True
             Me.FMT_Data.Visible = True
             Me.[SOF Forms/Reports].Visible = True
             Me.[PRR Selection/Status].Visible = True

        Else
             Me.[PRR Selection/Status].Visible = True
             Me.[PRR Forms/Reports].Visible = True
             Me.Admin.Visible = False
             Me.SOF_Interface.Visible = False
             Me.FMT_Data.Visible = False
             Me.[SOF Forms/Reports].Visible = False
        End If
    End With
    Dim fyVal As String
If IsNull(DLookup("[fy]", "tbl_fy", "[current_fy] = true")) Then
fyVal = ""
Else
fyVal = DLookup("[fy]", "tbl_fy", "[current_fy] = true")
End If

'Me.FY_IN.SetFocus
'Debug.Print "FY Value is " & fyVal
'Me.FY_IN.Text = fyVal
```

```
'Debug.Print "FY Received"
End Sub

Private Sub RolloverFY_Click()
    DoCmd.Close acForm, "frm_MainControl"
    DoCmd.OpenForm "frm_Closeout", acNormal
End Sub

Private Sub SOF_Emails_Click()
    DoCmd.OpenForm "frm_Emails"
    DoCmd.Close acForm, "frm_MainControl"
End Sub

Private Sub SOF_Update()
Application.Echo False

Call updateStatusBar("Updating SOFs and Director Dashboard.....")
Call updateSOF_Eng
Call clearStatusBar
Application.Echo True

End Sub
Private Sub SOFengineBackup()
Application.Echo False

Call updateStatusBar("Backing up the SOF Engine.....")
Call BEBackupNewSOF
Call clearStatusBar
Application.Echo True

End Sub
Private Sub UFR_Update()
Application.Echo False

Call updateStatusBar("Updating UFR data to the Director Dashboard.....")
Call UFRdashboard
Call clearStatusBar
Application.Echo True

End Sub
Private Sub Publish_Report()
Application.Echo False

Call updateStatusBar("Publishing Monthly Budget Report.....")
Call exportBudgetRpt
Call clearStatusBar
Application.Echo True

End Sub
```

```
Option Compare Database

Private Sub btnSubmit_Click()

    Dim strSQL As String
    Dim answer As Integer
    Dim runQuery As Boolean

    runQuery = True

    DoCmd.SetWarnings False

    If Me.WBS.Value = "" Or IsNull(Me.WBS.Value) Then

        MsgBox "Please select a Cost Collector before you try to submit a change."

        runQuery = False

    ElseIf Me.NewFunctionalArea.Value = "" Or IsNull(Me.NewFunctionalArea) Then

        MsgBox "Please select an New Functional Area before you try to submit a change."

        runQuery = False

    ElseIf Me.OldFunctionalArea.Value = "" Or IsNull(Me.OldFunctionalArea) Then

        answer = MsgBox("This Cost Collector does not have a Functional Area associated with it. Are y
ou sure you want to make this update?", vbQuestion + vbYesNo + vbDefaultButton2)

        If answer = vbNo Then runQuery = False

    End If

    If runQuery Then

        newFA = Me.NewFunctionalArea.Value
        newSAG = Left(Me.NewFunctionalArea.Value, 3)
        newAMSCO = Left(Me.NewFunctionalArea.Value, Len(Me.NewFunctionalArea.Value) - 4)
        newMDEP = Right(Me.NewFunctionalArea.Value, 4)

        strSQL = "UPDATE tbl_Requirements_and_Program SET [Functional Area] = """ & newFA & """, [SAG]
 = """ & newSAG & """, [AMSCO] = """ & newAMSCO & """, [MDEP] = """ & newMDEP & """" & _
        " WHERE [Cost Collector] = """ & Me.WBS.Value & """"

        DoCmd.RunSQL strSQL

        strSQL = "UPDATE tbl_PgmAndReqMonthly_Input SET [FunctionalArea] = """ & newFA & """ WHERE [WB
S] = """ & Me.WBS.Value & """"

        DoCmd.RunSQL strSQL

        strSQL = "UPDATE tbl_ProgramMonthly_Input SET [FunctionalArea] = """ & newFA & """ WHERE [WBS]
 = """ & Me.WBS.Value & """"

        DoCmd.RunSQL strSQL

        strSQL = "UPDATE tbl_RequirementOnly_Input SET [FunctionalAreaTo] = """ & newFA & """ WHERE [W
BS] = """ & Me.WBS.Value & """"

        DoCmd.RunSQL strSQL

        MsgBox "Update complete"

    End If

    DoCmd.SetWarnings False

End Sub

Private Sub WBS_AfterUpdate()

    Dim rst As DAO.Recordset
    Dim strSQL As String
    If Me.WBS.Value = "" Or IsNull(Me.WBS.Value) Then

        Me.OldFunctionalArea.Value = ""
        Exit Sub

    End If

    'Set the values for the Funding Source, AMO #, Functional Area, Fence Name, and Fenced WBS based o
n the cost collector that is chosen
    strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE [02_tbl_CostCollectorInformation].[C
ost Collector]= ""W/" & Me.WBS.Value & _
                                                            """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""O/" & Me.WBS.Value & _
                                                            """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""C/" & Me.WBS.Value & """"

    Set rst = CurrentDb.OpenRecordset(strSQL)

    'Me.FundingSource.Value = rst![Funding Source]
    'Me.FundingSource.Locked = True

    'Me.AMO_Packet.Locked = True
    'Me.AMO_Number.Locked = True

    'If IsNull(rst![AMO Number]) Or rst![AMO Number] = "" Then
    '    Me.AMO_Number.Visible = False
    '    Me.AMO_Number.Value = Null
    '    Me.AMO_Packet.Value = "No"
    'Else
    '    Me.AMO_Number.Visible = True
    '    Me.AMO_Number.Value = rst![AMO Number]
    '    Me.AMO_Packet.Value = "Yes"
    'End If

    Me.OldFunctionalArea.Value = rst![Functional Area]

    'Me.FencedName.Locked = True
    'Me.FencedName.Value = rst![Fenced]

    'Me.FencedWBS.Locked = True
    'Me.FencedWBS.Value = Me.WBS.Value

    rst.Close
    Set rst = Nothing

End Sub
```

```
Option Compare Database

Private Sub Form_Close()
DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.OpenForm "frm_MainControl"
    Call exportParentOrg
End Sub
```

```
Option Compare Database

Private Sub AMO_Packet_AfterUpdate()
DoCmd.SetWarnings False
    If Me.[AMO Packet] = "No" Then
        Me.AMO_Number.Visible = False
    ElseIf Me.[AMO Packet] = "Yes" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "N/A" Then
        Me.AMO_Number.Visible = False
    End If
DoCmd.SetWarnings True
End Sub

Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

    Call programInput

End Sub

Private Sub Command171_Click()

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalArea.Value = "" O
r Me.ObjectClass.Value = "" Or Me.FencedName.Value = "" Or Me.FencedWBS.Value = "" Or Me.Remarks.Value
 = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalArea) Or _
        IsNull(Me.ObjectClass) Or _
        IsNull(Me.FencedName) Or _
        IsNull(Me.FencedWBS) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Either delete the record to proceed or fill in the appropriate fields.", vbOKOnly, "Field Req
uired Notice"
    Else  'if required fields are NOT blank; input analyst name if blank

        Select Case programCheck
        Case 1

            Me.AnalystName.Value = Environ("USERNAME")
            Me.RequestDTG = Now() 'insert current date/time as the date/time of the request

            If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_P
gmAndReqMonthly_Input") & "-" & FYINFO(Now, "Y") & "PR"
            'Me.SetFocus
            Me.Dirty = False
            'DoCmd.RunCommand acCmdSaveRecord  'saves the record
            Call addPgmReqFolder
            Me.SetFocus
            DoCmd.OutputTo acOutputReport, "rpt_PgmAndReqFMT_Form", acFormatPDF, "C:\ProgramRequests\"
 & Me.RID.Value & ".pdf"  'saves a pdf the unassigned/unapproved copy of the request
            Call requestNoticePgmReqFMT 'calls fucntion to generate email notification for the request
            newRequest = MsgBox("Your request has been saved. Would you like to create another request
?", vbYesNo, "Request Notice")
            If newRequest = vbYes Then
                DoCmd.Close acForm, "frm_PgmAndReqFMT_Request"
                DoCmd.OpenForm "frm_PgmAndReqFMT_Request", acNormal
            Else
                Application.Quit
            End If
        'If the user input more program than the current budget for the Parent Org/Functional Area com
bo
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly and resubmit or delete the record. Contact your Branch Chief with any issues.", vbOKOnly, "Too
 Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
        End Select
    End If

End Sub

Private Sub Command254_Click()
    If Me.Dirty = True Then

        If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_PgmAn
dReqMonthly_Input") & "-" & FYINFO(Now, "Y") & "PR"

    End If

    DoCmd.OpenForm "frm_ActionNotice"
On Error GoTo errHandler:
    DoCmd.Close acForm, "frm_PgmAndReqFMT_Request"

errHandler:
    DoCmd.Close acForm, "frm_PgmAndReqFMT_Request"

End Sub

Private Sub Command357_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acLast
        Call programInput
    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."
End Sub

Private Sub Command358_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acFirst
            Call programInput
        Else
            DoCmd.GoToRecord , , acNext
            Call programInput
        End If

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command359_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acLast
            Call programInput
        Else
            DoCmd.GoToRecord , , acPrevious
            Call programInput
        End If

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command361_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acNewRec
        Call programInput
    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command363_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acFirst
        Call programInput
    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub ExtendedOrg_AfterUpdate()
        Me.Fund.Value = ""
        Me.Fund.Requery
        Me.FunctionalArea.Value = ""
        Me.FunctionalArea.Requery
        Me.FencedName.Value = ""
        Me.FencedName.Requery
        Me.FencedWBS.Value = ""
        Me.FencedWBS.Requery
        Me.WBS.Requery
        Me.WBS.Value = ""
End Sub

Private Sub Form_Load()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub FunctionalArea_AfterUpdate()

    Call programInput

End Sub

Private Sub FundingCategory_AfterUpdate()

    Call programInput

End Sub

Private Sub ObjectClass_AfterUpdate()
    If Me.ObjectClass = "25 - Contracts" Then
        MsgBox "Please be advised, you have selected the 'Contracts' object class which may require an
 AMO Number.", vbOKOnly, "Contract Selection Notice"
    End If
End Sub

Private Sub ParentOrg_AfterUpdate()
DoCmd.SetWarnings False
    Me.ExtendedOrg.Value = ""
    Me.ExtendedOrg.Requery
    Me.FunctionalArea.Value = ""
    Me.FunctionalArea.Requery
    Me.FencedName.Value = ""
    Me.FencedName.Requery
    Me.FencedWBS.Value = ""
    Me.FencedWBS.Requery
    Me.WBS.Requery
    Me.WBS.Value = ""
DoCmd.SetWarnings True
End Sub

Private Sub Text182_Click()
    Me.AnalystName = Environ("USERNAME")
End Sub

Private Sub TotalChangeTo_AfterUpdate()

    Select Case programCheck
        Case 1
            'Do nothing
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly before submitting or delete the record. Contact your Branch Chief with any issues.", vbOKOnly,
 "Too Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
    End Select

End Sub
```

```
Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
    Dim oFS As FileSystemObject
    Dim sPath As String

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalArea.Value = "" O
r Me.ObjectClass.Value = "" Or Me.FencedName.Value = "" Or Me.FencedWBS.Value = "" Or Me.Remarks.Value
 = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalArea) Or _
        IsNull(Me.ObjectClass) Or _
        IsNull(Me.FencedName) Or _
        IsNull(Me.FencedWBS) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Please fill in the appropriate fields before adding attachments.", vbOKOnly, "Field Required
Notice"
    Else  'if required fields are NOT blank; input analyst name if blank

        Select Case programCheck
        Case 1

            Me.AnalystName.Value = Environ("USERNAME")
            Me.RequestDTG = Now() 'insert current date/time as the date/time of the request

            If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_P
gmAndReqMonthly_Input") & "-" & FYINFO(Now, "Y") & "PR"
            'Me.SetFocus
            Me.Dirty = False
            'DoCmd.RunCommand acCmdSaveRecord  'saves the record

            Me.SetFocus
            sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Da
tabase\RequestAttachments\" & RID & " (FY" & FYINFO(Now, "Y") & ")\"
            Set oFS = New FileSystemObject

            If oFS.FolderExists(sPath) Then
            Else
                Call oFS.CreateFolder(sPath)

            End If

            CreateObject("Shell.Application").Open (sPath)

        'If the user input more program than the current budget for the Parent Org/Functional Area com
bo
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly and resubmit or delete the record. Contact your Branch Chief with any issues.", vbOKOnly, "Too
 Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
        End Select
    End If
End Sub

Private Sub Text408_LostFocus()

    'October Text Box
    If IsNull(Me.Text408.Value) Then
        Me.Text408.Value = 0
    End If

End Sub

Private Sub Text410_LostFocus()

    'November Text Box
    If IsNull(Me.Text410.Value) Then
        Me.Text410.Value = 0
    End If

End Sub

Private Sub Text412_LostFocus()

    'December Text Box
    If IsNull(Me.Text412.Value) Then
        Me.Text412.Value = 0
    End If

End Sub

Private Sub Text414_LostFocus()

    'January Text Box
    If IsNull(Me.Text414.Value) Then
        Me.Text414.Value = 0
    End If

End Sub

Private Sub Text416_LostFocus()

    'February Text Box
    If IsNull(Me.Text416.Value) Then
        Me.Text416.Value = 0
    End If

End Sub

Private Sub Text418_LostFocus()

    'March Text Box
    If IsNull(Me.Text418.Value) Then
        Me.Text418.Value = 0
    End If

End Sub

Private Sub Text436_LostFocus()

    'April Text Box
    If IsNull(Me.Text436.Value) Then
        Me.Text436.Value = 0
    End If

End Sub

Private Sub Text438_LostFocus()

    'May Text Box
    If IsNull(Me.Text438.Value) Then
        Me.Text438.Value = 0
    End If
End Sub

Private Sub Text440_LostFocus()

    'June Text Box
    If IsNull(Me.Text440.Value) Then
        Me.Text440.Value = 0
    End If

End Sub

Private Sub Text442_LostFocus()

    'July Text Box
    If IsNull(Me.Text442.Value) Then
        Me.Text442.Value = 0
    End If

End Sub

Private Sub Text444_LostFocus()

    'August Text Box
    If IsNull(Me.Text444.Value) Then
        Me.Text444.Value = 0
    End If

End Sub

Private Sub Text446_LostFocus()

    'September Text Box
    If IsNull(Me.Text446.Value) Then
        Me.Text446.Value = 0
    End If

End Sub

Private Sub Text408_Click()

    'October Text Box
    If Me.Text408.Value = 0 Then
        Me.Text408.Value = Null
    End If

End Sub

Private Sub Text410_Click()

    'November Text Box
    If Me.Text410.Value = 0 Then
        Me.Text410.Value = Null
    End If

End Sub

Private Sub Text412_Click()

    'December Text Box
    If Me.Text412.Value = 0 Then
        Me.Text412.Value = Null
    End If

End Sub

Private Sub Text414_Click()

    'January Text Box
    If Me.Text414.Value = 0 Then
        Me.Text414.Value = Null
    End If

End Sub
```

```
Private Sub Text416_Click()

    'February Text Box
    If Me.Text416.Value = 0 Then
        Me.Text416.Value = Null
    End If

End Sub

Private Sub Text418_Click()

    'March Text Box
    If Me.Text418.Value = 0 Then
        Me.Text418.Value = Null
    End If

End Sub

Private Sub Text436_Click()

    'April Text Box
    If Me.Text436.Value = 0 Then
        Me.Text436.Value = Null
    End If

End Sub

Private Sub Text438_Click()

    'May Text Box
    If Me.Text438.Value = 0 Then
        Me.Text438.Value = Null
    End If

End Sub

Private Sub Text440_Click()

    'June Text Box
    If Me.Text440.Value = 0 Then
        Me.Text440.Value = Null
    End If

End Sub

Private Sub Text442_Click()

    'July Text Box
    If Me.Text442.Value = 0 Then
        Me.Text442.Value = Null
    End If

End Sub

Private Sub Text444_Click()

    'August Text Box
    If Me.Text444.Value = 0 Then
        Me.Text444.Value = Null
    End If

End Sub

Private Sub Text446_Click()

    'September Text Box
    If Me.Text446.Value = 0 Then
        Me.Text446.Value = Null
    End If

End Sub
```

```
Private Sub WBS_AfterUpdate()

    Dim rst As DAO.Recordset
    Dim strSQL As String

    'Set the values for the Funding Source, AMO #, Functional Area, Fence Name, and Fenced WBS based o
n the cost collector that is chosen
    strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE RIGHT([02_tbl_CostCollectorInformati
on].[Cost Collector], LEN([02_tbl_CostCollectorInformation].[Cost Collector]) -2)=""" & Me.WBS.Value &
 """"

    Set rst = CurrentDb.OpenRecordset(strSQL)

    Me.FundingSource.Value = "FMT"
    Me.FundingSource.Locked = True

    Me.AMO_Packet.Locked = True
    Me.AMO_Number.Locked = True

    If IsNull(rst![AMO Number]) Or rst![AMO Number] = "" Then
        Me.AMO_Number.Visible = False
        Me.AMO_Number.Value = Null
        Me.AMO_Packet.Value = "No"
    Else
        Me.AMO_Number.Visible = True
        Me.AMO_Number.Value = rst![AMO Number]
        Me.AMO_Packet.Value = "Yes"
    End If

    If IsNull(rst![Functional Area]) Or rst![Functional Area] = "" Then
        Me.FunctionalArea.Locked = False
    Else
        Me.FunctionalArea.Locked = True
    End If

    Me.FunctionalArea.Value = rst![Functional Area]

    Me.FencedName.Locked = True
    Me.FencedName.Value = rst![Fenced]

    Me.FencedWBS.Locked = True
    Me.FencedWBS.Value = Me.WBS.Value

    Me.Contract_Name.Value = rst![Contract Name]
    Me.Contract_Name.Locked = True

    Call programInput

    rst.Close
    Set rst = Nothing

End Sub

Private Sub programInput()

    Dim rst As DAO.Recordset
    Dim strSQL As String

    cContractName = Me.[Contract_Name]

    'String added to make the SQL statement properly look for Contract Names if there is one otherwise
 looks for NULL values (IS NULL)
    If cContractName = "" Or IsNull(cContractName) Then

        strAddSQL = "IS NULL"

    Else

        strAddSQL = "= """ & cContractName & """"

    End If

    strSQL = "SELECT SUM([Unit Program]) AS Limit FROM [Unit Funding] WHERE [Parent Organization] = ""
```

```
" & Me.ParentOrg & _
                                                                     """ And [Functional Area] = """ &
Me.FunctionalArea & _
                                                                     """ And [Fund] = """ & Me.Fund & _
                                                                     """ And [Contract Name] " & strAdd
SQL

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!Limit) Then

        programSQL = 0
        Me.txtPgmTotal.Value = 0

    Else

        programSQL = rst!Limit
        Me.txtPgmTotal.Value = programSQL

    End If

    totalAdj = 0

    'PgmOnly + From
    strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_ProgramMonthly_Input WHERE ParentOrg = """ & M
e.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
    & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ AND (Contract_Name " &
strAddSQL & " OR Contract_Name = """ & cContractName & """)"
    Set rst = CurrentDb.OpenRecordset(strSQL)
    If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

    'PgmReq + From
    strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_PgmAndReqMonthly_Input WHERE ParentOrg = """ &
 Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
    & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ AND (Contract_Name " &
strAddSQL & " OR Contract_Name = """ & cContractName & """)"
    Set rst = CurrentDb.OpenRecordset(strSQL)
    If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

    Me.txtPgmUsed.Value = totalAdj
    Me.txtPgmRemaining.Value = programSQL - totalAdj

End Sub

Function programCheck() As Integer

    Dim rst As DAO.Recordset
    Dim strSQL As String

    If Me.TotalPgm.Value <= 0 Then

        programCheck = 1
        Exit Function

    End If

    cContractName = Me.[Contract_Name]

    'String added to make the SQL statement properly look for Contract Names if there is one otherwise
 looks for NULL values (IS NULL)
    If cContractName = "" Or IsNull(cContractName) Then

        strAddSQL = "IS NULL"

    Else

        strAddSQL = "= """ & cContractName & """"

    End If

    strSQL = "SELECT SUM([Unit Program]) AS Limit FROM [Unit Funding] WHERE [Parent Organization] = ""
" & Me.ParentOrg & _
                                                                     """ And [Functional Area] = """ &
```

```
Me.FunctionalArea & _
                                                                     """ And [Fund] = """ & Me.Fund & _
                                                                     """ And [Contract Name] " & strAdd
SQL

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!Limit) Then

        programCheck = -1

    Else
        programSQL = rst!Limit

        totalAdj = 0

        'PgmReq + From
        strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_PgmAndReqMonthly_Input WHERE ParentOrg = "
"" & Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
        & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ AND (Contract_Name
" & strAddSQL & " OR Contract_Name = """ & cContractName & """)"
        Set rst = CurrentDb.OpenRecordset(strSQL)
        If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

        'PgmOnly + From
        strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_ProgramMonthly_Input WHERE ParentOrg = """
 & Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
        & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ AND (Contract_Name
" & strAddSQL & " OR Contract_Name = """ & cContractName & """)"
        Set rst = CurrentDb.OpenRecordset(strSQL)
        If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

        totalAdj = totalAdj + Me.TotalPgm.Value

        If programSQL >= totalAdj Then

            programCheck = 1

        Else

            programCheck = 0

        End If

    End If

    rst.Close
    Set rst = Nothing

End Function
```

```
Option Compare Database

Private Sub AMO_Packet_AfterUpdate()
DoCmd.SetWarnings False
    If Me.[AMO Packet] = "No" Then
        Me.AMO_Number.Visible = False
    ElseIf Me.[AMO Packet] = "Yes" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "N/A" Then
        Me.AMO_Number.Visible = False
    End If
DoCmd.SetWarnings True
End Sub

Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

    Call programInput

End Sub

Private Sub Command171_Click()

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalArea.Value = "" O
r Me.ObjectClass.Value = "" Or Me.FencedName.Value = "" Or Me.FencedWBS.Value = "" Or Me.Remarks.Value
 = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalArea) Or _
        IsNull(Me.ObjectClass) Or _
        IsNull(Me.FencedName) Or _
        IsNull(Me.FencedWBS) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Either delete the record to proceed or fill in the appropriate fields.", vbOKOnly, "Field Req
uired Notice"
    Else  'if required fields are NOT blank; input analyst name if blank

        Select Case programCheck
        Case 1

            Me.AnalystName.Value = Environ("USERNAME")
            Me.RequestDTG = Now() 'insert current date/time as the date/time of the request

            If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_P
gmAndReqMonthly_Input") & "-" & FYINFO(Now, "Y") & "PR"
            'Me.SetFocus
            Me.Dirty = False
            'DoCmd.RunCommand acCmdSaveRecord  'saves the record
            Call addPgmReqFolder
            Me.SetFocus
            DoCmd.OutputTo acOutputReport, "rpt_PgmAndReqMonthly_Form", acFormatPDF, "C:\ProgramReques
ts\" & Me.RID.Value & ".pdf"  'saves a pdf the unassigned/unapproved copy of the request
            Call requestNoticePgmReqMon 'calls fucntion to generate email notification for the request
            newRequest = MsgBox("Your request has been saved. Would you like to create another request
?", vbYesNo, "Request Notice")
            If newRequest = vbYes Then
                DoCmd.OpenForm "frm_TypeRequest", acNormal
                DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request"
            Else
                Application.Quit
            End If
        'If the user input more program than the current budget for the Parent Org/Functional Area com
bo
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly and resubmit or delete the record. Contact your Branch Chief with any issues.", vbOKOnly, "Too
 Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
        End Select
    End If

End Sub

Private Sub Command254_Click()
    If Me.Dirty = True Then

        If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_PgmAn
dReqMonthly_Input") & "-" & FYINFO(Now, "Y") & "PR"

    End If

    DoCmd.OpenForm "frm_ActionNotice"
On Error GoTo errHandler:
    DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request"

errHandler:
    DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request"

End Sub

Private Sub Command357_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acLast
        Call programInput
    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."
End Sub

Private Sub Command358_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acFirst
            Call programInput
        Else
            DoCmd.GoToRecord , , acNext
            Call programInput
        End If

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command359_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acLast
            Call programInput
        Else
            DoCmd.GoToRecord , , acPrevious
            Call programInput
        End If

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command361_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acNewRec
        Call programInput
    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command363_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acFirst
        Call programInput
    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub ExtendedOrg_AfterUpdate()
        Me.Fund.Value = ""
        Me.Fund.Requery
        Me.FunctionalArea.Value = ""
        Me.FunctionalArea.Requery
        Me.FencedName.Value = ""
        Me.FencedName.Requery
        Me.FencedWBS.Value = ""
        Me.FencedWBS.Requery
        Me.WBS.Requery
        Me.WBS.Value = ""
End Sub

Private Sub Form_Load()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub FunctionalArea_AfterUpdate()

    Call programInput

End Sub

Private Sub FundingCategory_AfterUpdate()

    Call programInput

End Sub

Private Sub ObjectClass_AfterUpdate()
    If Me.ObjectClass = "25 - Contracts" Then
        MsgBox "Please be advised, you have selected the 'Contracts' object class which may require an
 AMO Number.", vbOKOnly, "Contract Selection Notice"
    End If
End Sub

Private Sub ParentOrg_AfterUpdate()
DoCmd.SetWarnings False
    Me.ExtendedOrg.Value = ""
    Me.ExtendedOrg.Requery
    Me.FunctionalArea.Value = ""
    Me.FunctionalArea.Requery
    Me.FencedName.Value = ""
    Me.FencedName.Requery
    Me.FencedWBS.Value = ""
    Me.FencedWBS.Requery
    Me.WBS.Requery
    Me.WBS.Value = ""
DoCmd.SetWarnings True
End Sub

Private Sub Text182_Click()
    Me.AnalystName = Environ("USERNAME")
End Sub

Private Sub TotalChangeTo_AfterUpdate()

    Select Case programCheck
        Case 1
            'Do nothing
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly before submitting or delete the record. Contact your Branch Chief with any issues.", vbOKOnly,
 "Too Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
    End Select

End Sub
```

```
Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
    Dim oFS As FileSystemObject
    Dim sPath As String

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalArea.Value = "" O
r Me.ObjectClass.Value = "" Or Me.FencedName.Value = "" Or Me.FencedWBS.Value = "" Or Me.Remarks.Value
 = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalArea) Or _
        IsNull(Me.ObjectClass) Or _
        IsNull(Me.FencedName) Or _
        IsNull(Me.FencedWBS) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Please fill in the appropriate fields before adding attachments.", vbOKOnly, "Field Required
Notice"
    Else  'if required fields are NOT blank; input analyst name if blank

        Select Case programCheck
        Case 1

            Me.AnalystName.Value = Environ("USERNAME")
            Me.RequestDTG = Now() 'insert current date/time as the date/time of the request

            If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_P
gmAndReqMonthly_Input") & "-" & FYINFO(Now, "Y") & "PR"
            'Me.SetFocus
            Me.Dirty = False
            'DoCmd.RunCommand acCmdSaveRecord  'saves the record

            Me.SetFocus
            sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Da
tabase\RequestAttachments\" & RID & " (FY" & FYINFO(Now, "Y") & ")\"
            Set oFS = New FileSystemObject

            If oFS.FolderExists(sPath) Then
            Else
                Call oFS.CreateFolder(sPath)

            End If

            CreateObject("Shell.Application").Open (sPath)

        'If the user input more program than the current budget for the Parent Org/Functional Area com
bo
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly and resubmit or delete the record. Contact your Branch Chief with any issues.", vbOKOnly, "Too
 Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
        End Select
    End If
End Sub

Private Sub Text408_LostFocus()

    'October Text Box
    If IsNull(Me.Text408.Value) Then
        Me.Text408.Value = 0
    End If

End Sub

Private Sub Text410_LostFocus()

    'November Text Box
    If IsNull(Me.Text410.Value) Then
        Me.Text410.Value = 0
    End If

End Sub

Private Sub Text412_LostFocus()

    'December Text Box
    If IsNull(Me.Text412.Value) Then
        Me.Text412.Value = 0
    End If

End Sub

Private Sub Text414_LostFocus()

    'January Text Box
    If IsNull(Me.Text414.Value) Then
        Me.Text414.Value = 0
    End If

End Sub

Private Sub Text416_LostFocus()

    'February Text Box
    If IsNull(Me.Text416.Value) Then
        Me.Text416.Value = 0
    End If

End Sub

Private Sub Text418_LostFocus()

    'March Text Box
    If IsNull(Me.Text418.Value) Then
        Me.Text418.Value = 0
    End If

End Sub

Private Sub Text436_LostFocus()

    'April Text Box
    If IsNull(Me.Text436.Value) Then
        Me.Text436.Value = 0
    End If

End Sub

Private Sub Text438_LostFocus()

    'May Text Box
    If IsNull(Me.Text438.Value) Then
        Me.Text438.Value = 0
    End If
End Sub

Private Sub Text440_LostFocus()

    'June Text Box
    If IsNull(Me.Text440.Value) Then
        Me.Text440.Value = 0
    End If

End Sub

Private Sub Text442_LostFocus()

    'July Text Box
    If IsNull(Me.Text442.Value) Then
        Me.Text442.Value = 0
    End If

End Sub

Private Sub Text444_LostFocus()

    'August Text Box
    If IsNull(Me.Text444.Value) Then
        Me.Text444.Value = 0
    End If

End Sub

Private Sub Text446_LostFocus()

    'September Text Box
    If IsNull(Me.Text446.Value) Then
        Me.Text446.Value = 0
    End If

End Sub

Private Sub Text408_Click()

    'October Text Box
    If Me.Text408.Value = 0 Then
        Me.Text408.Value = Null
    End If

End Sub

Private Sub Text410_Click()

    'November Text Box
    If Me.Text410.Value = 0 Then
        Me.Text410.Value = Null
    End If

End Sub

Private Sub Text412_Click()

    'December Text Box
    If Me.Text412.Value = 0 Then
        Me.Text412.Value = Null
    End If

End Sub

Private Sub Text414_Click()

    'January Text Box
    If Me.Text414.Value = 0 Then
        Me.Text414.Value = Null
    End If

End Sub
```

```
Private Sub Text416_Click()

    'February Text Box
    If Me.Text416.Value = 0 Then
        Me.Text416.Value = Null
    End If

End Sub

Private Sub Text418_Click()

    'March Text Box
    If Me.Text418.Value = 0 Then
        Me.Text418.Value = Null
    End If

End Sub

Private Sub Text436_Click()

    'April Text Box
    If Me.Text436.Value = 0 Then
        Me.Text436.Value = Null
    End If

End Sub

Private Sub Text438_Click()

    'May Text Box
    If Me.Text438.Value = 0 Then
        Me.Text438.Value = Null
    End If

End Sub

Private Sub Text440_Click()

    'June Text Box
    If Me.Text440.Value = 0 Then
        Me.Text440.Value = Null
    End If

End Sub

Private Sub Text442_Click()

    'July Text Box
    If Me.Text442.Value = 0 Then
        Me.Text442.Value = Null
    End If

End Sub

Private Sub Text444_Click()

    'August Text Box
    If Me.Text444.Value = 0 Then
        Me.Text444.Value = Null
    End If

End Sub

Private Sub Text446_Click()

    'September Text Box
    If Me.Text446.Value = 0 Then
        Me.Text446.Value = Null
    End If

End Sub
```

```
Private Sub WBS_AfterUpdate()

    Dim rst As DAO.Recordset
    Dim strSQL As String

    'Set the values for the Funding Source, AMO #, Functional Area, Fence Name, and Fenced WBS based o
n the cost collector that is chosen
    strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE [02_tbl_CostCollectorInformation].[C
ost Collector]= ""W/" & Me.WBS.Value & _
                                                          """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""O/" & Me.WBS.Value & _
                                                          """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""C/" & Me.WBS.Value & """"

    'Not used because it caused a weird issue specifically with Cost Collector "S.0060288.3.9.8.3.1"
    'strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE RIGHT([02_tbl_CostCollectorInformat
ion].[Cost Collector], LEN([02_tbl_CostCollectorInformation].[Cost Collector]) -2)=""" & Me.WBS.Value
& """"

    Set rst = CurrentDb.OpenRecordset(strSQL)

    Me.FundingSource.Value = rst![Funding Source]
    Me.FundingSource.Locked = True

    Me.AMO_Packet.Locked = True
    Me.AMO_Number.Locked = True

    If IsNull(rst![AMO Number]) Or rst![AMO Number] = "" Then
        Me.AMO_Number.Visible = False
        Me.AMO_Number.Value = Null
        Me.AMO_Packet.Value = "No"
    Else
        Me.AMO_Number.Visible = True
        Me.AMO_Number.Value = rst![AMO Number]
        Me.AMO_Packet.Value = "Yes"
    End If

    If IsNull(rst![Functional Area]) Or rst![Functional Area] = "" Then
        Me.FunctionalArea.Locked = False
    Else
        Me.FunctionalArea.Locked = True
    End If

    Me.FunctionalArea.Value = rst![Functional Area]

    Me.FencedName.Locked = True
    Me.FencedName.Value = rst![Fenced]

    Me.FencedWBS.Locked = True
    Me.FencedWBS.Value = Me.WBS.Value

    Me.Contract_Name.Value = rst![Contract Name]
    Me.Contract_Name.Locked = True

    Call programInput

    rst.Close
    Set rst = Nothing

End Sub

Private Sub programInput()

    Dim rst As DAO.Recordset
    Dim strSQL As String

    cContractName = Me.[Contract_Name]

    'String added to make the SQL statement properly look for Contract Names if there is one otherwise
 looks for NULL values (IS NULL)
    If cContractName = "" Or IsNull(cContractName) Then

        strAddSQL = "IS NULL"
    Else

        strAddSQL = "= """ & cContractName & """"

    End If

    strSQL = "SELECT SUM([Unit Program]) AS Limit FROM [Unit Funding] WHERE [Parent Organization] = ""
" & Me.ParentOrg & _
                                                                    """ And [Functional Area] = """ &
Me.FunctionalArea & _
                                                                    """ And [Fund] = """ & Me.Fund & _
                                                                    """ And [Funding Category] = """ &
 Me.FundingCategory & _
                                                                    """ And [Contract Name] " & strAdd
SQL

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!Limit) Then

        programSQL = 0
        Me.txtPgmTotal.Value = 0

    Else

        programSQL = rst!Limit
        Me.txtPgmTotal.Value = programSQL

    End If

    totalAdj = 0

    'PgmOnly + From
    strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_ProgramMonthly_Input WHERE ParentOrg = """ & M
e.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
    & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategory] =
 """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & _
    " OR Contract_Name = """ & cContractName & """)"
    Set rst = CurrentDb.OpenRecordset(strSQL)
    If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

    'PgmReq + From
    strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_PgmAndReqMonthly_Input WHERE ParentOrg = """ &
 Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
    & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategory] =
 """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & _
    " OR Contract_Name = """ & cContractName & """)"
    Set rst = CurrentDb.OpenRecordset(strSQL)
    If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

    Me.txtPgmUsed.Value = totalAdj
    Me.txtPgmRemaining.Value = programSQL - totalAdj

End Sub

Function programCheck() As Integer

    Dim rst As DAO.Recordset
    Dim strSQL As String

    If Me.TotalPgm.Value <= 0 Then

        programCheck = 1
        Exit Function

    End If

    cContractName = Me.[Contract_Name]

    'String added to make the SQL statement properly look for Contract Names if there is one otherwise
 looks for NULL values (IS NULL)
    If cContractName = "" Or IsNull(cContractName) Then
```

```
        strAddSQL = "IS NULL"

    Else

        strAddSQL = "= """ & cContractName & """"

    End If

    strSQL = "SELECT SUM([Unit Program]) AS Limit FROM [Unit Funding] WHERE [Parent Organization] = ""
" & Me.ParentOrg & _
                                                                    """ And [Functional Area] = """ &
Me.FunctionalArea & _
                                                                    """ And [Fund] = """ & Me.Fund & _
                                                                    """ And [Funding Category] = """ &
 Me.FundingCategory & _
                                                                    """ And [Contract Name] " & strAdd
SQL

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!Limit) Then

        programCheck = -1

    Else
        programSQL = rst!Limit

        totalAdj = 0

        'PgmReq + From
        strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_ProgramMonthly_Input WHERE ParentOrg = """
 & Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
        & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategor
y] = """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & _
        " OR Contract_Name = """ & cContractName & """)"
        Set rst = CurrentDb.OpenRecordset(strSQL)
        If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

        'PgmOnly + From
        strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_PgmAndReqMonthly_Input WHERE ParentOrg = "
"" & Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
        & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategor
y] = """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & _
        " OR Contract_Name = """ & cContractName & """)"
        Set rst = CurrentDb.OpenRecordset(strSQL)
        If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

        totalAdj = totalAdj + Me.TotalPgm.Value

        If programSQL >= totalAdj Then

            programCheck = 1

        Else

            programCheck = 0

        End If

    End If

    rst.Close
    Set rst = Nothing

End Function
```

```
Option Compare Database


Private Sub BC_Name_GotFocus()
    Me.BC_Name.Value = Environ("USERNAME")
End Sub



Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()
DoCmd.SetWarnings False

    If Me.Frame223 = 0 Then
        approvalMsg = MsgBox("Are you sure you want to approve this record?", vbYesNo)
On Error GoTo errHandler:
            If approvalMsg = vbYes Then
                Me.Frame223.Value = -1
                If IsNull(Me.BC_Comments) Then
                    Me.BC_Comments.Value = "APPROVED"
                Else
                    Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
                End If
                Me.BC_Name.Value = Environ("USERNAME")
                Me.BC_ApprovedDTG = Now()
                Me.Dirty = False
                Call BCApprovalPgmReqMon
                Me.Requery
                addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
                    If addRequests = vbYes Then
                        DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BC"
                        'DoCmd.GoToRecord , , acNext
                    Else
                        DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request_BC"
                        DoCmd.OpenForm "frm_Transition"
                    End If
            Else
                Me.Refresh
            End If
    ElseIf Me.Frame223 = -1 Then
        approvalMsg = MsgBox("Are you sure you want to approve this record?", vbYesNo)
On Error GoTo errHandler:
            If approvalMsg = vbYes Then
                If IsNull(Me.BC_Comments) Then
                    Me.BC_Comments.Value = "APPROVED"
                Else
                    Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
                End If
                Me.BC_Name.Value = Environ("USERNAME")
                Me.BC_ApprovedDTG = Now()
                Me.Dirty = False
                Call BCApprovalPgmReqMon
                Me.Requery
                addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
                    If addRequests = vbYes Then
                        DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BC"
                        'DoCmd.GoToRecord , , acNext
                    Else
```

```
                        DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request_BC"
                        DoCmd.OpenForm "frm_Transition"
                    End If
            Else
                Me.Refresh
            End If
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Refresh
    End If
End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False

    If Me.Frame223 = 0 And IsNull(Me.BC_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.Frame223 = 0 And Not IsNull(Me.BC_Comments) Then
        Me.BC_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BC_RejectedDTG) Then
            Me.BC_RejectedDTG = Now()
        'Else
            'Me.BC_RejectedDTG.Value = Me.BC_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "REJECTED"
        Me.Dirty = False
        Call rejectNoticeBCPgmReqMon
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
                DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BC"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BC_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request_BC"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!FencedWBS.Requery
    Me.FencedWBS.Value = ""
    Me!FencedName.Requery
    Me.FencedName.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me!ExtendedOrg.Requery
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!FencedWBS.Requery
    Me.FencedWBS.Value = ""
    Me!FencedName.Requery
    Me.FencedName.Value = ""
End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "F (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database


Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()

    DoCmd.SetWarnings False
    Me.BC_Approved.Value = -1
    Me.BC_Name.Value = Environ("USERNAME")
    Me.BC_ApprovedDTG = Now()
    If IsNull(Me.BC_Comments) Then
        Me.BC_Comments.Value = "APPROVED"
    Else
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
    End If
    Me.Dirty = False
    DoCmd.OutputTo acOutputReport, "rpt_PgmAndReqMonthlyBCSub_Form", acFormatPDF, "\\RUCKW0U9G67001\dr
m\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\RequestBackups\" & Me.RID.Value &
 ".pdf"
    Call emailBCSubPRMon

    If Left(Me.Fund.Value, 7) <> "202010F" Then

        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery

    ElseIf Me.Fund.Value Like "202010F*" Then
        Call byMonthQueries 'Runs just the normal queries
        Me.Form.Requery

    Else
        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery
    End If

    addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
    If addRequests = vbYes Then
        DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BC_Sub"
        Me.Form.Requery
    Else
        DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request_BC_Sub"
        DoCmd.OpenForm "frm_Transition"
    End If

    DoCmd.SetWarnings True

End Sub

Private Sub byMonthQueriesWH()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.NovemberPgm.Value <> 0 Then
```

```
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.AugustPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCWH"
    End If

    DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCR"

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub byMonthQueries()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.NovemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.AugustPgm.Value <> 0 Then
```

```
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCP"
    End If

    DoCmd.OpenQuery "qry_PgmAndReqMonthlyBCR"

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False
    If Me.BC_Approved = 0 And IsNull(Me.BC_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.BC_Approved = 0 And Not IsNull(Me.BC_Comments) Then
        Me.BC_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BO_RejectedDTG) Then
            Me.BC_RejectedDTG = Now()
        'Else
            'Me.BO_RejectedDTG.Value = Me.BO_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "REJECTED"
        Me.Dirty = False
        Call emailBCSubRejPRMon
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
                DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BC_Sub"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BO_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request_BC_Sub"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me!ExtendedOrg.Requery
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""

End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "F (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database


Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()

    DoCmd.SetWarnings False
    Me.BO_Approved.Value = -1
    Me.BO_Name.Value = Environ("USERNAME")
    Me.BO_ApprovedDTG = Now()
    If IsNull(Me.BO_Comments) Then
        Me.BO_Comments.Value = "APPROVED"
    Else
        Me.BO_Comments.Value = Me.BO_Comments.Value & vbCrLf & "APPROVED"
    End If
    Me.Dirty = False
    DoCmd.OutputTo acOutputReport, "rpt_PgmAndReqMonthlyBU_Form", acFormatPDF, "\\RUCKW0U9G67001\drm\P
BD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\RequestBackups\" & Me.RID.Value & ".
pdf"
    Call BOApprovalPgmReqMon

    If Left(Me.Fund.Value, 7) <> "202010F" Then

        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery

    ElseIf Me.Fund.Value Like "202010F*" Then
        Call byMonthQueries 'Runs just the normal queries
        Me.Form.Requery

    Else
        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery
    End If

    addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
    If addRequests = vbYes Then
        DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BO"
        Me.Form.Requery
    Else
        DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request_BO"
        DoCmd.OpenForm "frm_Transition"
    End If

    DoCmd.SetWarnings True

End Sub

Private Sub byMonthQueriesWH()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.NovemberPgm.Value <> 0 Then
```

```
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.AugustPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyWH"
    End If

    DoCmd.OpenQuery "qry_PgmAndReqMonthlyR"

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub byMonthQueries()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.NovemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.AugustPgm.Value <> 0 Then
```

```
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmAndReqMonthlyP"
    End If

    DoCmd.OpenQuery "qry_PgmAndReqMonthlyR"

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False
    If Me.BO_Approved = 0 And IsNull(Me.BO_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.BO_Approved = 0 And Not IsNull(Me.BO_Comments) Then
        Me.BO_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BO_RejectedDTG) Then
            Me.BO_RejectedDTG = Now()
        'Else
            'Me.BO_RejectedDTG.Value = Me.BO_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Approved.Value = 0
        Me.BO_Comments.Value = Me.BO_Comments.Value & vbCrLf & "REJECTED"
        Me.Dirty = False
        Call rejectNoticeBOPgmReqMon
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
                DoCmd.OpenForm "frm_PgmAndReqMonthly_Request_BO"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BO_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_PgmAndReqMonthly_Request_BO"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me!ExtendedOrg.Requery
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""

End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "F (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database

Private Sub Frame0_Click()
    If Frame0 = 1 Then
        DoCmd.OpenForm "frm_TypeRequest"
        DoCmd.Close acForm, "frm_MainControl"
    ElseIf Frame0 = 2 Then
        DoCmd.OpenForm "frm_BCSub_UnapprovedCount"
        DoCmd.Close acForm, "frm_MainControl"
    'Removed BO from approval process. Keepiong code for now in case current/future change their minds
.
    'Else
        'DoCmd.OpenForm "frm_BO_UnapprovedCount"
        'DoCmd.Close acForm, "frm_MainControl"
    End If
End Sub
```

```
Option Compare Database

Private Sub AMO_Packet_AfterUpdate()
DoCmd.SetWarnings False
    If Me.[AMO Packet] = "No" Then
        Me.AMO_Number.Visible = False
    ElseIf Me.[AMO Packet] = "Yes" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "N/A" Then
        Me.AMO_Number.Visible = False
    End If
DoCmd.SetWarnings True
End Sub

Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

        Call programInput

    End If

End Sub

Private Sub Command171_Click()

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalArea.Value = "" O
r Me.ObjectClass.Value = "" Or Me.FencedName.Value = "" Or Me.FencedWBS.Value = "" Or Me.Remarks.Value
 = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalArea) Or _
        IsNull(Me.ObjectClass) Or _
        IsNull(Me.FencedName) Or _
        IsNull(Me.FencedWBS) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Either delete the record to proceed or fill in the appropriate fields.", vbOKOnly, "Field Req
uired Notice"
    Else  'if required fields are NOT blank; input analyst name if blank

        Select Case programCheck
        Case 1

            Me.AnalystName.Value = Environ("USERNAME")
            Me.RequestDTG = Now() 'insert current date/time as the date/time of the request

            If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_P
rogramMonthly_Input") & "-" & FYINFO(Now, "Y") & "P"
            'Me.SetFocus
            Me.Dirty = False
            'DoCmd.RunCommand acCmdSaveRecord  'saves the record
            Call addPgmReqFolder
            Me.SetFocus
            DoCmd.OutputTo acOutputReport, "rpt_ProgramMonthly_Form", acFormatPDF, "C:\ProgramRequests
\" & Me.RID.Value & ".pdf"  'saves a pdf the unassigned/unapproved copy of the request
            Call requestNoticePgmMon 'calls fucntion to generate email notification for the request
            newRequest = MsgBox("Your request has been saved. Would you like to create another request
?", vbYesNo, "Request Notice")
            If newRequest = vbYes Then
                DoCmd.OpenForm "frm_TypeRequest", acNormal
                DoCmd.Close acForm, "frm_ProgramMonthly_Request"
            Else
                Application.Quit
            End If
        'If the user input more program than the current budget for the Parent Org/Functional Area com
bo
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly and resubmit or delete the record. Contact your Branch Chief with any issues.", vbOKOnly, "Too
 Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
        End Select
    End If

End Sub

Private Sub Command254_Click()
    If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_ProgramMo
nthly_Input") & "-" & FYINFO(Now, "Y") & "P"
    DoCmd.OpenForm "frm_ActionNotice"
On Error GoTo errHandler:
    DoCmd.Close acForm, "frm_ProgramMonthly_Request"

errHandler:
    DoCmd.Close acForm, "frm_ProgramMonthly_Request"

End Sub

Private Sub Command357_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acLast
        Call programInput
    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."
End Sub

Private Sub Command358_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acFirst
        Else
            DoCmd.GoToRecord , , acNext
        End If
        Call programInput

    End If
```

```
Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command359_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acLast
        Else
            DoCmd.GoToRecord , , acPrevious
        End If
        Call programInput

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command361_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acNewRec

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command363_Click()

    If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else
        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acFirst
        Call programInput

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub ExtendedOrg_AfterUpdate()
        Me.Fund.Value = ""
        Me.Fund.Requery
        Me.FunctionalArea.Value = ""
        Me.FunctionalArea.Requery
        Me.FencedName.Value = ""
        Me.FencedName.Requery
        Me.FencedWBS.Value = ""
        Me.FencedWBS.Requery
        Me.WBS.Requery
        Me.WBS.Value = ""
End Sub

Private Sub Form_Load()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub FunctionalArea_AfterUpdate()

    Call programInput

End Sub

Private Sub FundingCategory_AfterUpdate()

    Call programInput

End Sub

Private Sub ObjectClass_AfterUpdate()
    If Me.ObjectClass = "25 - Contracts" Then
        MsgBox "Please be advised, you have selected the 'Contracts' object class which may require an
 AMO Number.", vbOKOnly, "Contract Selection Notice"
    End If
End Sub

Private Sub ParentOrg_AfterUpdate()
DoCmd.SetWarnings False
    Me.ExtendedOrg.Value = ""
    Me.ExtendedOrg.Requery
    Me.FunctionalArea.Value = ""
    Me.FunctionalArea.Requery
    Me.FencedName.Value = ""
    Me.FencedName.Requery
    Me.FencedWBS.Value = ""
    Me.FencedWBS.Requery
    Me.WBS.Requery
    Me.WBS.Value = ""
DoCmd.SetWarnings True
End Sub

Private Sub Text182_Click()
    Me.AnalystName = Environ("USERNAME")
End Sub

Private Sub TotalChangeTo_AfterUpdate()

    Select Case programCheck
        Case 1
            'Do nothing
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly before submitting or delete the record. Contact your Branch Chief with any issues.", vbOKOnly,
 "Too Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
    End Select

End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
```

```
Dim oFS As FileSystemObject
Dim sPath As String

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalArea.Value = "" O
r Me.ObjectClass.Value = "" Or Me.FencedName.Value = "" Or Me.FencedWBS.Value = "" Or Me.Remarks.Value
 = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalArea) Or _
        IsNull(Me.ObjectClass) Or _
        IsNull(Me.FencedName) Or _
        IsNull(Me.FencedWBS) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Please fill in the appropriate fields before adding attachments.", vbOKOnly, "Field Required
Notice"
    Else  'if required fields are NOT blank; input analyst name if blank

        Select Case programCheck
        Case 1

            Me.AnalystName.Value = Environ("USERNAME")
            Me.RequestDTG = Now() 'insert current date/time as the date/time of the request

            If Me.Text348.Value = "" Or IsNull(Me.Text348.Value) Then Me.Text348.Value = addRID("tbl_P
rogramMonthly_Input") & "-" & FYINFO(Now, "Y") & "P"
            'Me.SetFocus
            Me.Dirty = False

            'DoCmd.RunCommand acCmdSaveRecord  'saves the record
            Me.SetFocus

            sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Da
tabase\RequestAttachments\" & RID & " (FY" & FYINFO(Now, "Y") & ")\"
            Set oFS = New FileSystemObject

            If oFS.FolderExists(sPath) Then
            Else
                Call oFS.CreateFolder(sPath)

            End If

            CreateObject("Shell.Application").Open (sPath)

        'If the user input more program than the current budget for the Parent Org/Functional Area com
bo
        Case 0
            MsgBox "You have entered too much program for this Parent Org/Functional Area. Adjust acco
rdingly and resubmit or delete the record. Contact your Branch Chief with any issues.", vbOKOnly, "Too
 Much Program"
        'If the ParentOrg/Functional area combo does not have a budget associated with it
        Case -1
            MsgBox "This Parent Org/Functional Area does not have a budget so no program may be alloca
ted. Change Parent Org/Functional Area or delete the record. Contact your Branch Chief with any issues
.", vbOKOnly, "No Program"
        Case Else
            MsgBox "Error submitting request. Please contact the database administrator."
        End Select
    End If


End Sub

Private Sub Text408_Click()
```

```
    'October Text Box
    If Me.Text408.Value = 0 Then
        Me.Text408.Value = Null
    End If

End Sub

Private Sub Text408_LostFocus()

    'October Text Box
    If IsNull(Me.Text408.Value) Then
        Me.Text408.Value = 0
    End If

End Sub

Private Sub Text410_LostFocus()

    'November Text Box
    If IsNull(Me.Text410.Value) Then
        Me.Text410.Value = 0
    End If

End Sub

Private Sub Text412_LostFocus()

    'December Text Box
    If IsNull(Me.Text412.Value) Then
        Me.Text412.Value = 0
    End If

End Sub

Private Sub Text414_LostFocus()

    'January Text Box
    If IsNull(Me.Text414.Value) Then
        Me.Text414.Value = 0
    End If

End Sub

Private Sub Text416_LostFocus()

    'February Text Box
    If IsNull(Me.Text416.Value) Then
        Me.Text416.Value = 0
    End If

End Sub

Private Sub Text418_LostFocus()

    'March Text Box
    If IsNull(Me.Text418.Value) Then
        Me.Text418.Value = 0
    End If

End Sub

Private Sub Text436_LostFocus()

    'April Text Box
    If IsNull(Me.Text436.Value) Then
        Me.Text436.Value = 0
    End If

End Sub

Private Sub Text438_LostFocus()
```

```
    'May Text Box
    If IsNull(Me.Text438.Value) Then
        Me.Text438.Value = 0
    End If

End Sub

Private Sub Text440_LostFocus()

    'June Text Box
    If IsNull(Me.Text440.Value) Then
        Me.Text440.Value = 0
    End If

End Sub

Private Sub Text442_LostFocus()

    'July Text Box
    If IsNull(Me.Text442.Value) Then
        Me.Text442.Value = 0
    End If

End Sub

Private Sub Text444_LostFocus()

    'August Text Box
    If IsNull(Me.Text444.Value) Then
        Me.Text444.Value = 0
    End If

End Sub

Private Sub Text446_LostFocus()

    'September Text Box
    If IsNull(Me.Text446.Value) Then
        Me.Text446.Value = 0
    End If

End Sub

Private Sub Text410_Click()

    'November Text Box
    If Me.Text410.Value = 0 Then
        Me.Text410.Value = Null
    End If

End Sub

Private Sub Text412_Click()

    'December Text Box
    If Me.Text412.Value = 0 Then
        Me.Text412.Value = Null
    End If

End Sub

Private Sub Text414_Click()

    'January Text Box
    If Me.Text414.Value = 0 Then
        Me.Text414.Value = Null
    End If

End Sub

Private Sub Text416_Click()

    'February Text Box
    If Me.Text416.Value = 0 Then
        Me.Text416.Value = Null
    End If

End Sub

Private Sub Text418_Click()

    'March Text Box
    If Me.Text418.Value = 0 Then
        Me.Text418.Value = Null
    End If

End Sub

Private Sub Text436_Click()

    'April Text Box
    If Me.Text436.Value = 0 Then
        Me.Text436.Value = Null
    End If

End Sub

Private Sub Text438_Click()

    'May Text Box
    If Me.Text438.Value = 0 Then
        Me.Text438.Value = Null
    End If

End Sub

Private Sub Text440_Click()

    'June Text Box
    If Me.Text440.Value = 0 Then
        Me.Text440.Value = Null
    End If

End Sub

Private Sub Text442_Click()

    'July Text Box
    If Me.Text442.Value = 0 Then
        Me.Text442.Value = Null
    End If

End Sub

Private Sub Text444_Click()

    'August Text Box
    If Me.Text444.Value = 0 Then
        Me.Text444.Value = Null
    End If

End Sub

Private Sub Text446_Click()

    'September Text Box
    If Me.Text446.Value = 0 Then
        Me.Text446.Value = Null
    End If

End Sub

Private Sub WBS_AfterUpdate()

    Dim rst As DAO.Recordset
    Dim strSQL As String
    'Set the values for the Funding Source, AMO #, Functional Area, Fence Name, and Fenced WBS based o
n the cost collector that is chosen
    strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE [02_tbl_CostCollectorInformation].[C
ost Collector]= ""W/" & Me.WBS.Value & _
                                                            """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""O/" & Me.WBS.Value & _
                                                            """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""C/" & Me.WBS.Value & """"

    'Not used because it caused a weird issue specifically with Cost Collector "S.0060288.3.9.8.3.1"
    'strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE RIGHT([02_tbl_CostCollectorInformat
ion].[Cost Collector], LEN([02_tbl_CostCollectorInformation].[Cost Collector]) -2)=""" & Me.WBS.Value
& """"

    Set rst = CurrentDb.OpenRecordset(strSQL)

    Me.FundingSource.Value = rst![Funding Source]
    Me.FundingSource.Locked = True

    Me.AMO_Packet.Locked = True
    Me.AMO_Number.Locked = True

    If IsNull(rst![AMO Number]) Or rst![AMO Number] = "" Then
        Me.AMO_Number.Visible = False
        Me.AMO_Number.Value = Null
        Me.AMO_Packet.Value = "No"
    Else
        Me.AMO_Number.Visible = True
        Me.AMO_Number.Value = rst![AMO Number]
        Me.AMO_Packet.Value = "Yes"
    End If

    If IsNull(rst![Functional Area]) Or rst![Functional Area] = "" Then
        Me.FunctionalArea.Locked = False
    Else
        Me.FunctionalArea.Locked = True
    End If

    Me.FunctionalArea.Value = rst![Functional Area]

    Me.FencedName.Locked = True
    Me.FencedName.Value = rst![Fenced]

    Me.FencedWBS.Locked = True
    Me.FencedWBS.Value = Me.WBS.Value

    Me.Contract_Name.Value = rst![Contract Name]
    Me.Contract_Name.Locked = True

    Call programInput

    rst.Close
    Set rst = Nothing

End Sub

Private Sub programInput()

    Dim rst As DAO.Recordset
    Dim strSQL As String

    cContractName = Me.[Contract_Name]

    'String added to make the SQL statement properly look for Contract Names if there is one otherwise
 looks for NULL values (IS NULL)
    If cContractName = "" Or IsNull(cContractName) Then

        strAddSQL = "IS NULL"

    Else

        strAddSQL = "= """ & cContractName & """"
    End If

    strSQL = "SELECT SUM([Unit Program]) AS Limit FROM [Unit Funding] WHERE [Parent Organization] = ""
" & Me.ParentOrg & _
                                                                    """ And [Functional Area] = """ &
Me.FunctionalArea & _
                                                                    """ And [Fund] = """ & Me.Fund & _
                                                                    """ And [Funding Category] = """ &
 Me.FundingCategory & _
                                                                    """ And [Contract Name] " & strAdd
SQL

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!Limit) Then

        programSQL = 0
        Me.txtPgmTotal.Value = 0

    Else

        programSQL = rst!Limit
        Me.txtPgmTotal.Value = programSQL

    End If

    totalAdj = 0

    'PgmOnly + From
    strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_ProgramMonthly_Input WHERE ParentOrg = """ & M
e.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
    & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategory] =
 """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & " OR Contract_Name = """ & cContrac
tName & """)"
    Set rst = CurrentDb.OpenRecordset(strSQL)
    If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

    'PgmReq + From
    strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_PgmAndReqMonthly_Input WHERE ParentOrg = """ &
 Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
    & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategory] =
 """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & " OR Contract_Name = """ & cContrac
tName & """)"
    Set rst = CurrentDb.OpenRecordset(strSQL)
    If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

    Me.txtPgmUsed.Value = totalAdj
    Me.txtPgmRemaining.Value = programSQL - totalAdj

End Sub

Function programCheck() As Integer

    Dim rst As DAO.Recordset
    Dim strSQL As String

    If Me.TotalPgm.Value <= 0 Then

        programCheck = 1
        Exit Function

    End If

    cContractName = Me.[Contract_Name]

    'String added to make the SQL statement properly look for Contract Names if there is one otherwise
 looks for NULL values (IS NULL)
    If cContractName = "" Or IsNull(cContractName) Then

        strAddSQL = "IS NULL"

    Else
```

```
        strAddSQL = "= """ & cContractName & """"

    End If

    strSQL = "SELECT SUM([Unit Program]) AS Limit FROM [Unit Funding] WHERE [Parent Organization] = ""
" & Me.ParentOrg & _
                                                                    """ And [Functional Area] = """ &
Me.FunctionalArea & _
                                                                    """ And [Fund] = """ & Me.Fund & _
                                                                    """ And [Funding Category] = """ &
 Me.FundingCategory & _
                                                                    """ And [Contract Name] " & strAdd
SQL

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!Limit) Then

        programCheck = -1

    Else
        programSQL = rst!Limit

        totalAdj = 0

        'PgmReq + From
        strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_PgmAndReqMonthly_Input WHERE ParentOrg = "
"" & Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
        & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategor
y] = """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & " OR Contract_Name = """ & cCon
tractName & """)"
        Set rst = CurrentDb.OpenRecordset(strSQL)
        If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

        'PgmOnly + From
        strSQL = "SELECT SUM(TotalPgm) AS Adjusted FROM tbl_ProgramMonthly_Input WHERE ParentOrg = """
 & Me.ParentOrg & """ And FunctionalArea = """ & Me.FunctionalArea & """" _
        & " And RID <> """ & Me.Text348.Value & """ And Fund = """ & Me.Fund & """ And [FundingCategor
y] = """ & Me.FundingCategory & """ AND (Contract_Name " & strAddSQL & " OR Contract_Name = """ & cCon
tractName & """)"
        Set rst = CurrentDb.OpenRecordset(strSQL)
        If Not IsNull(rst!Adjusted) Then totalAdj = totalAdj + rst!Adjusted

        totalAdj = totalAdj + Me.TotalPgm.Value

        If programSQL >= totalAdj Then

            programCheck = 1

        Else

            programCheck = 0

        End If

    End If

    rst.Close
    Set rst = Nothing

End Function
```

```
Option Compare Database


Private Sub BC_Name_GotFocus()
    Me.BC_Name.Value = Environ("USERNAME")
End Sub



Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()
DoCmd.SetWarnings False

    If Me.Frame223 = 0 Then
        approvalMsg = MsgBox("Are you sure you want to approve this record?", vbYesNo)
On Error GoTo errHandler:
            If approvalMsg = vbYes Then
                Me.Frame223.Value = -1
                If IsNull(Me.BC_Comments) Then
                    Me.BC_Comments.Value = "APPROVED"
                Else
                    Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
                End If
                Me.BC_Name.Value = Environ("USERNAME")
                Me.BC_ApprovedDTG = Now()
                Me.Dirty = False
                Call BCApprovalPgmMon
                Me.Requery
                addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
                    If addRequests = vbYes Then
                        DoCmd.OpenForm "frm_ProgramMonthly_Request_BC"
                        'DoCmd.GoToRecord , , acNext
                    Else
                        DoCmd.Close acForm, "frm_ProgramMonthly_Request_BC"
                        DoCmd.OpenForm "frm_Transition"
                    End If
            Else
                Me.Refresh
            End If
    ElseIf Me.Frame223 = -1 Then
        approvalMsg = MsgBox("Are you sure you want to approve this record?", vbYesNo)
On Error GoTo errHandler:
            If approvalMsg = vbYes Then
                If IsNull(Me.BC_Comments) Then
                    Me.BC_Comments.Value = "APPROVED"
                Else
                    Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
                End If
                Me.BC_Name.Value = Environ("USERNAME")
                Me.BC_ApprovedDTG = Now()
                Me.Dirty = False
                Call BCApprovalPgmMon
                Me.Requery
                addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
                    If addRequests = vbYes Then
                        DoCmd.OpenForm "frm_ProgramMonthly_Request_BC"
                        'DoCmd.GoToRecord , , acNext
                    Else
```

```
                        DoCmd.Close acForm, "frm_ProgramMonthly_Request_BC"
                        DoCmd.OpenForm "frm_Transition"
                    End If
            Else
                Me.Refresh
            End If
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Refresh
    End If
End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False

    If Me.Frame223 = 0 And IsNull(Me.BC_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.Frame223 = 0 And Not IsNull(Me.BC_Comments) Then
        Me.BC_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BC_RejectedDTG) Then
            Me.BC_RejectedDTG = Now()
        'Else
            'Me.BC_RejectedDTG.Value = Me.BC_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "REJECTED"
        Me.Dirty = False
        Call rejectNoticeBCPgmMon
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
                DoCmd.OpenForm "frm_ProgramMonthly_Request_BC"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BC_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_ProgramMonthly_Request_BC"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!FencedWBS.Requery
    Me.FencedWBS.Value = ""
    Me!FencedName.Requery
    Me.FencedName.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""
End Sub


Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me!ExtendedOrg.Requery
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!FencedWBS.Requery
    Me.FencedWBS.Value = ""
    Me!FencedName.Requery
    Me.FencedName.Value = ""
End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "F (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database


Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()

    DoCmd.SetWarnings False
    Me.BC_Approved.Value = -1
    Me.BC_Name.Value = Environ("USERNAME")
    Me.BC_ApprovedDTG = Now()
    If IsNull(Me.BC_Comments) Then
        Me.BC_Comments.Value = "APPROVED"
    Else
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
    End If
    Me.Dirty = False
    DoCmd.OutputTo acOutputReport, "rpt_ProgramMonthlyBC_Form", acFormatPDF, "\\RUCKW0U9G67001\drm\PBD
\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\RequestBackups\" & Me.RID.Value & ".pd
f"
    Call emailBCSubPgm

    If Left(Me.Fund.Value, 7) <> "202010F" Then

        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery

    ElseIf Me.Fund.Value Like "202010F*" Then
        Call byMonthQueries 'Runs just the normal queries
        Me.Form.Requery

    Else
        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery
    End If

    addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
    If addRequests = vbYes Then
        DoCmd.OpenForm "frm_ProgramMonthly_Request_BC_Sub"
        Me.Form.Requery
    Else
        DoCmd.Close acForm, "frm_ProgramMonthly_Request_BC_Sub"
        DoCmd.OpenForm "frm_Transition"
    End If

    DoCmd.SetWarnings True

End Sub

Private Sub byMonthQueriesWH()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.NovemberPgm.Value <> 0 Then
```

```
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.AugustPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
        DoCmd.OpenQuery "qry_PgmMonthlyBCWH"
    End If

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub byMonthQueries()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.NovemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.AugustPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthlyBC"
    End If

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False
    If Me.BC_Approved = 0 And IsNull(Me.BC_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.BC_Approved = 0 And Not IsNull(Me.BC_Comments) Then
        Me.BC_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BC_RejectedDTG) Then
            Me.BC_RejectedDTG = Now()
        'Else
            'Me.BC_RejectedDTG.Value = Me.BC_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Approved.Value = 0
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "REJECTED"
        Me.Dirty = False
        Call emailBCSubRejPgm
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
                DoCmd.OpenForm "frm_ProgramMonthly_Request_BC_Sub"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BC_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_ProgramMonthly_Request_BC_Sub"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me!ExtendedOrg.Requery
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""

End Sub
```

```
Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "F (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database


Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()

    DoCmd.SetWarnings False
    Me.BO_Approved.Value = -1
    Me.BO_Name.Value = Environ("USERNAME")
    Me.BO_ApprovedDTG = Now()
    If IsNull(Me.BO_Comments) Then
        Me.BO_Comments.Value = "APPROVED"
    Else
        Me.BO_Comments.Value = Me.BO_Comments.Value & vbCrLf & "APPROVED"
    End If
    Me.Dirty = False
    DoCmd.OutputTo acOutputReport, "rpt_ProgramMonthlyBU_Form", acFormatPDF, "\\RUCKW0U9G67001\drm\PBD
\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\RequestBackups\" & Me.RID.Value & ".pd
f"
    Call BOApprovalPgmMon

    If Left(Me.Fund.Value, 7) <> "202010F" Then

        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery

    ElseIf Me.Fund.Value Like "202010F*" Then
        Call byMonthQueries 'Runs just the normal queries
        Me.Form.Requery

    Else
        Call byMonthQueriesWH 'Runs both the normal and WH queries
        Me.Form.Requery
    End If

    addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
    If addRequests = vbYes Then
        DoCmd.OpenForm "frm_ProgramMonthly_Request_BO"
        Me.Form.Requery
    Else
        DoCmd.Close acForm, "frm_ProgramMonthly_Request_BO"
        DoCmd.OpenForm "frm_Transition"
    End If

    DoCmd.SetWarnings True

End Sub

Private Sub byMonthQueriesWH()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.NovemberPgm.Value <> 0 Then
```

```
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.AugustPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
        DoCmd.OpenQuery "qry_PgmMonthlyWH"
    End If

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub byMonthQueries()

    If Me.OctoberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "October"
        Me.MonthAmount.Value = Me.OctoberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.NovemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "November"
        Me.MonthAmount.Value = Me.NovemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.DecemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "December"
        Me.MonthAmount.Value = Me.DecemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.JanuaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "January"
        Me.MonthAmount.Value = Me.JanuaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.FebruaryPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "February"
        Me.MonthAmount.Value = Me.FebruaryPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.MarchPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "March"
        Me.MonthAmount.Value = Me.MarchPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.AprilPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "April"
        Me.MonthAmount.Value = Me.AprilPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.MayPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "May"
        Me.MonthAmount.Value = Me.MayPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.JunePgm.Value <> 0 Then
        Me.FiscalMonth.Value = "June"
        Me.MonthAmount.Value = Me.JunePgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.JulyPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "July"
        Me.MonthAmount.Value = Me.JulyPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.AugustPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "August"
        Me.MonthAmount.Value = Me.AugustPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    If Me.SeptemberPgm.Value <> 0 Then
        Me.FiscalMonth.Value = "September"
        Me.MonthAmount.Value = Me.SeptemberPgm.Value
        DoCmd.OpenQuery "qry_PgmMonthly"
    End If

    Me.FiscalMonth.Value = ""
    Me.MonthAmount.Value = 0

End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False
    If Me.BO_Approved = 0 And IsNull(Me.BO_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.BO_Approved = 0 And Not IsNull(Me.BO_Comments) Then
        Me.BO_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BO_RejectedDTG) Then
            Me.BO_RejectedDTG = Now()
        'Else
            'Me.BO_RejectedDTG.Value = Me.BO_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Approved.Value = 0
        Me.BO_Comments.Value = Me.BO_Comments.Value & vbCrLf & "REJECTED"
        Me.Dirty = False
        Call rejectNoticeBOPgmMon
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
                DoCmd.OpenForm "frm_ProgramMonthly_Request_BO"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BO_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_ProgramMonthly_Request_BO"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me!ExtendedOrg.Requery
    Me!FunctionalArea.Requery
    Me.FunctionalArea.Value = ""
    Me!WBS.Requery
    Me.WBS.Value = ""

End Sub
```

```
Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "F (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database

Private Sub btn_PRRStatusDefault_Click()

    Call setDefaults

End Sub

Private Sub btn_PRRStatusRefresh_Click()

    frm_PRRStatusSubForm.Requery

    Dim rst As DAO.Recordset
    Dim strSQL As String

    strFC = [Forms]![frm_PRRStatus]![cmb_PPRStatusFC]
    strFund = [Forms]![frm_PRRStatus]![cmb_PPRStatusFund]
    strPO = [Forms]![frm_PRRStatus]![cmb_PPRStatusPO]
    strFA = [Forms]![frm_PRRStatus]![cmb_PPRStatusFA]
    strCN = [Forms]![frm_PRRStatus]![cmb_PPRStatusCN]

    strSQL = "SELECT SUM(qry_PRRStatusSort.TotalPgm) As PgmTtl From qry_PRRStatusSort " _
    & "WHERE qry_PRRStatusSort.FundCenter LIKE '" & strFC & "' AND qry_PRRStatusSort.Fund LIKE '" & st
rFund & "' " _
    & "AND qry_PRRStatusSort.ParentOrg LIKE '" & strPO & "' AND qry_PRRStatusSort.FunctionalArea LIKE
'" & strFA & "' " _
    & "AND (qry_PRRStatusSort.Contract_Name LIKE IIF(ISNULL('" & strCN & "'), '', '" & strCN & "') OR
" _
    & "IIF('" & strCN & "' = '*', ISNULL(qry_PRRStatusSort.Contract_Name), qry_PRRStatusSort.Contract_
Name LIKE '" & strCN & "'))"

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!pgmTtl) Then

        txt_PgmTotal.Value = 0

    Else

        txt_PgmTotal.Value = rst!pgmTtl

    End If

    If strCN = "" Or IsNull(strCN) Then

        strAddSQL = "[Contract Name] IS NULL"

    ElseIf strCN = "*" Then

        strAddSQL = "([Contract Name] LIKE ""*"" OR ISNULL([Contract Name]))"

    Else

        strAddSQL = "[Contract Name] LIKE """ & strCN & """"

    End If

    strSQL = "SELECT SUM([Unit Program]) AS Limit FROM [Unit Funding] WHERE [Parent Organization] LIKE
 """ & strPO & _
                                                                """ And [Functional Area] LIKE """ & s
trFA & _
                                                                """ And [Fund Center] LIKE """ & strFC
 & _
                                                                """ And [Fund] Like """ & strFund & _
                                                                """ And " & strAddSQL

    Set rst = CurrentDb.OpenRecordset(strSQL)

    If IsNull(rst!Limit) Then

        txt_PgmLimit.Value = 0

    Else
```

```
        txt_PgmLimit.Value = rst!Limit

    End If

    txt_PgmRemain.Value = txt_PgmLimit.Value - txt_PgmTotal.Value

End Sub

Sub setDefaults()

    cmb_PPRStatusFC.Value = "*"
    cmb_PPRStatusFund.Value = "*"
    cmb_PPRStatusPO.Value = "*"
    cmb_PPRStatusFA.Value = "*"
    cmb_PPRStatusCN.Value = "*"

    frm_PRRStatusSubForm.Requery

    txt_PgmTotal.Value = 0
    txt_PgmLimit.Value = 0
    txt_PgmRemain.Value = 0

End Sub

Private Sub Form_Load()

    Call setDefaults

End Sub
```

```
Option Compare Database




Private Sub Cost_Collector__To__AfterUpdate()
Dim dbs As Database, rst As Recordset
Dim SQL As String
Set dbs = CurrentDb

SQL = "SELECT [Functional Area] FROM [02_tbl_CostCollectorInformation]" & _
        "WHERE [02_tbl_CostCollectorInformation].[Cost Collector] = '" & Me.[Cost Collector (To)] & "'
"

If Me.[Cost Collector (To)] = "" Then
        Me.[Functional Area (To)] = ""
    Else
        Set rst = dbs.OpenRecordset(SQL)
        Me.[Functional Area (To)] = rst![Functional Area]
        '= rst![SQL]
        Me.Refresh
End If

End Sub

Private Sub Form_Close()
DoCmd.SetWarnings (WarningsOff)
    'DoCmd.OpenQuery "qry_Delete_Neg_Fenced"
    DoCmd.OpenQuery "000_Delete_FMT_Neg_Fenced"
    DoCmd.OpenQuery "qry_Append_Neg_Fenced"
    Call exportFMTlog
End Sub
```

```
Option Compare Database

Private Sub btn_PRRStatusRefresh_Click()

    Me.Requery

End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.Close , "frm_Requests_ALL"
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.Close , "frm_Requests_Archived"
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub AMO_Packet_AfterUpdate()
DoCmd.SetWarnings False
    If Me.[AMO Packet] = "No" Then
        Me.AMO_Number.Visible = False
    ElseIf Me.[AMO Packet] = "Yes" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "" Then
        Me.AMO_Number.Visible = True
    ElseIf Me.[AMO Packet] = "N/A" Then
        Me.AMO_Number.Visible = False
    End If
DoCmd.SetWarnings True
End Sub

Private Sub Combo302_AfterUpdate()

If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

    DoCmd.GoToRecord , , acLast

    If Me.Text361.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text361.Value = Me.Combo302.Value

    End If

End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."
End Sub

Private Sub Command171_Click()

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalAreaTo.Value = ""
 Or Me.ObjectClassTo.Value = "" Or Me.FencedNameTo.Value = "" Or Me.FencedWBSTo.Value = "" Or _
        Me.AmountTo.Value = "" Or Me.TotalChangeTo.Value = "" Or Me.NewAmountTo.Value = "" Or Me.Remar
ks.Value = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalAreaTo) Or _
        IsNull(Me.ObjectClassTo) Or _
        IsNull(Me.FencedNameTo) Or _
        IsNull(Me.FencedWBSTo) Or _
        IsNull(Me.AmountTo) Or _
        IsNull(Me.TotalChangeTo) Or _
        IsNull(Me.NewAmountTo) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Either delete the record to proceed or fill in the appropriate fields.", vbOKOnly, "Field Req
uired Notice"
    Else
        Me.AnalystName.Value = Environ("USERNAME")
        'If IsNull(Me.RequestDTG) Then
```

```
            Me.RequestDTG = Now()
        'Else
            'Me.RequestDTG.Value = Me.RequestDTG.Value & vbCrLf & Now()
        'End If

        If Me.Text361.Value = "" Or IsNull(Me.Text361.Value) Then Me.Text361.Value = addRID("tbl_Requi
rementOnly_Input") & "-" & FYINFO(Now, "Y") & "R"

        Me.Dirty = False
        'DoCmd.RunCommand acCmdSaveRecord
        Call addPgmReqFolder
        DoCmd.OutputTo acOutputReport, "rpt_RequirementOnly_Form", acFormatPDF, "C:\ProgramRequests\"
& Me.RID.Value & ".pdf"
        Call requestNoticeReq
        newRequest = MsgBox("Your request has been saved. Would you like to create another request?",
vbYesNo, "Request Notice")
            If newRequest = vbYes Then
                DoCmd.OpenForm "frm_TypeRequest", acNormal
                DoCmd.Close acForm, "frm_RequirementOnly_Request"
            Else
                Application.Quit
            End If
    End If

End Sub

Private Sub Command254_Click()
    If Me.Text361.Value = "" Or IsNull(Me.Text361.Value) Then Me.Text361.Value = addRID("tbl_Requireme
ntOnly_Input") & "-" & FYINFO(Now, "Y") & "R"
    DoCmd.OpenForm "frm_ActionNotice"
On Error GoTo errHandler:
    DoCmd.Close acForm, "frm_RequirementOnly_Request"

errHandler:
    DoCmd.Close acForm, "frm_RequirementOnly_Request"

End Sub

Private Sub Command357_Click()

If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acLast

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command358_Click()

If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acFirst
        Else
            DoCmd.GoToRecord , , acNext
        End If

    End If

Exit Sub
```

```
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command359_Click()

If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        If Me.RID.Value = "" Or IsNull(Me.RID.Value) Then
            DoCmd.GoToRecord , , acLast
        Else
            DoCmd.GoToRecord , , acPrevious
        End If

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command361_Click()

If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acNewRec

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub Command363_Click()

If Me.Dirty = True Then
        MsgBox "You must fill in fields with gray labels. Either delete the record to proceed or fill
in the appropriate fields.", vbOKOnly, "Field Required Notice"
    Else

        On Error GoTo ErrorHandler
        DoCmd.GoToRecord , , acFirst

    End If

Exit Sub
ErrorHandler:
    MsgBox "There are no records to naviagte to."

End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me.Fund.Value = ""
    Me.Fund.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.FunctionalAreaTo.Requery
    Me.FencedNameTo.Value = ""
    Me.FencedNameTo.Requery
    Me.FencedWBSTo.Value = ""
    Me.FencedWBSTo.Requery
    Me.WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub Form_Load()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub ObjectClassTo_AfterUpdate()
    If Me.ObjectClassTo = "25 - Contracts" Then
        MsgBox "Please be advised, you have selected the 'Contracts' object class which may require an
 AMO Number.", vbOKOnly, "Contract Selection Notice"
    End If
End Sub

Private Sub ParentOrg_AfterUpdate()
DoCmd.SetWarnings False
    Me.ExtendedOrg.Value = ""
    Me.ExtendedOrg.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.FunctionalAreaTo.Requery
    Me.FencedNameTo.Value = ""
    Me.FencedNameTo.Requery
    Me.FencedWBSTo.Value = ""
    Me.FencedWBSTo.Requery
    Me.WBS.Requery
    Me.WBS.Value = ""
DoCmd.SetWarnings True
End Sub

Private Sub Text182_Click()
    Me.AnalystName = Environ("USERNAME")
End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    If Me.ParentOrg.Value = "" Or Me.ExtendedOrg.Value = "" Or Me.FundCenter.Value = "" Or Me.Fund.Val
ue = "" Or Me.Reason.Value = "" Or Me.WBS.Value = "" Or Me.FundingSource.Value = "" Or _
        Me.AMO_Packet.Value = "" Or Me.MissionCriticality.Value = "" Or Me.FunctionalAreaTo.Value = ""
 Or Me.ObjectClassTo.Value = "" Or Me.FencedNameTo.Value = "" Or Me.FencedWBSTo.Value = "" Or _
        Me.AmountTo.Value = "" Or Me.TotalChangeTo.Value = "" Or Me.NewAmountTo.Value = "" Or Me.Remar
ks.Value = "" Or _
        IsNull(Me.ParentOrg) Or _
        IsNull(Me.ExtendedOrg) Or _
        IsNull(Me.FundCenter) Or _
        IsNull(Me.Fund) Or _
        IsNull(Me.Reason) Or _
        IsNull(Me.WBS) Or _
        IsNull(Me.FundingSource) Or _
        IsNull(Me.AMO_Packet) Or _
        IsNull(Me.MissionCriticality) Or _
        IsNull(Me.FunctionalAreaTo) Or _
        IsNull(Me.ObjectClassTo) Or _
        IsNull(Me.FencedNameTo) Or _
        IsNull(Me.FencedWBSTo) Or _
        IsNull(Me.AmountTo) Or _
        IsNull(Me.TotalChangeTo) Or _
        IsNull(Me.NewAmountTo) Or _
        IsNull(Me.Remarks) Then
            MsgBox "You must provide a justification for this request and fill in all fields with gray
 labels. Please fill in the appropriate fields before adding attachments.", vbOKOnly, "Field Required
Notice"
    Else
        Me.AnalystName.Value = Environ("USERNAME")
        'If IsNull(Me.RequestDTG) Then
            Me.RequestDTG = Now()
        'Else
```

```
            'Me.RequestDTG.Value = Me.RequestDTG.Value & vbCrLf & Now()
        'End If

        If Me.Text361.Value = "" Or IsNull(Me.Text361.Value) Then Me.Text361.Value = addRID("tbl_Requi
rementOnly_Input") & "-" & FYINFO(Now, "Y") & "R"

        Me.Dirty = False

        sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Databa
se\RequestAttachments\" & RID & " (FY" & FYINFO(Now, "Y") & ")\"
        Set oFS = New FileSystemObject

        If oFS.FolderExists(sPath) Then
        Else
            Call oFS.CreateFolder(sPath)

        End If

        CreateObject("Shell.Application").Open (sPath)

    End If


End Sub

Private Sub WBS_AfterUpdate()

    Dim rst As DAO.Recordset
    Dim strSQL As String

    'Set the values for the Funding Source, AMO #, Functional Area, Fence Name, and Fenced WBS based o
n the cost collector that is chosen
    strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE [02_tbl_CostCollectorInformation].[C
ost Collector]= ""W/" & Me.WBS.Value & _
                                                            """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""O/" & Me.WBS.Value & _
                                                            """ OR [02_tbl_CostCollectorInformation].[C
ost Collector]= ""C/" & Me.WBS.Value & """"

    'Not used because it caused a weird issue specifically with Cost Collector "S.0060288.3.9.8.3.1"
    'strSQL = "SELECT * FROM 02_tbl_CostCollectorInformation WHERE RIGHT([02_tbl_CostCollectorInformat
ion].[Cost Collector], LEN([02_tbl_CostCollectorInformation].[Cost Collector]) -2)=""" & Me.WBS.Value
& """"

    Set rst = CurrentDb.OpenRecordset(strSQL)

    Me.FundingSource.Value = rst![Funding Source]
    Me.FundingSource.Locked = True

    Me.AMO_Packet.Locked = True
    Me.AMO_Number.Locked = True

    If IsNull(rst![AMO Number]) Or rst![AMO Number] = "" Then
        Me.AMO_Number.Visible = False
        Me.AMO_Number.Value = Null
        Me.AMO_Packet.Value = "No"
    Else
        Me.AMO_Number.Visible = True
        Me.AMO_Number.Value = rst![AMO Number]
        Me.AMO_Packet.Value = "Yes"
    End If

    If IsNull(rst![Functional Area]) Or rst![Functional Area] = "" Then

        Me.FunctionalAreaTo.Locked = False
        Me.FunctionalAreaTo.Value = ""

    Else

        Me.FunctionalAreaTo.Locked = True
        Me.FunctionalAreaTo.Value = rst![Functional Area]

    End If
    Me.FencedNameTo.Locked = True
    Me.FencedNameTo.Value = rst![Fenced]

    Me.FencedWBSTo.Locked = True
    Me.FencedWBSTo.Value = Me.WBS.Value

    rst.Close
    Set rst = Nothing

End Sub
```

```
Option Compare Database


Private Sub BC_Name_GotFocus()
    Me.BC_Name.Value = Environ("USERNAME")
End Sub



Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()
DoCmd.SetWarnings False

    If Me.Frame223 = 0 Then
        approvalMsg = MsgBox("Are you sure you want to approve this record?", vbYesNo)
On Error GoTo errHandler:
            If approvalMsg = vbYes Then
                Me.Frame223.Value = -1
                If IsNull(Me.BC_Comments) Then
                    Me.BC_Comments.Value = "APPROVED"
                Else
                    Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
                End If
                Me.BC_Name.Value = Environ("USERNAME")
                Me.BC_ApprovedDTG = Now()
                Me.Dirty = False
                Call BCApprovalReq
                'Me.Requery
                addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
                    If addRequests = vbYes Then
                        DoCmd.OpenForm "frm_RequirementOnly_Request_BC"
                        'DoCmd.GoToRecord , , acNext
                    Else
                        DoCmd.Close acForm, "frm_RequirementOnly_Request_BC"
                        DoCmd.OpenForm "frm_Transition"
                    End If
            Else
                Me.Refresh
            End If
    ElseIf Me.Frame223 = -1 Then
        approvalMsg = MsgBox("Are you sure you want to approve this record?", vbYesNo)
On Error GoTo errHandler:
            If approvalMsg = vbYes Then
                If IsNull(Me.BC_Comments) Then
                    Me.BC_Comments.Value = "APPROVED"
                Else
                    Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
                End If
                Me.BC_Name.Value = Environ("USERNAME")
                Me.BC_ApprovedDTG = Now()
                Me.Dirty = False
                Call BCApprovalReq
                'Me.Requery
                addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
                    If addRequests = vbYes Then
                        DoCmd.OpenForm "frm_RequirementOnly_Request_BC"
                        'DoCmd.GoToRecord , , acNext
                    Else
```

```
                        DoCmd.Close acForm, "frm_RequirementOnly_Request_BC"
                        DoCmd.OpenForm "frm_Transition"
                    End If
            Else
                Me.Refresh
            End If
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Refresh
    End If
End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False

    If Me.Frame223 = 0 And IsNull(Me.BC_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.Frame223 = 0 And Not IsNull(Me.BC_Comments) Then
        Me.BC_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BC_RejectedDTG) Then
            Me.BC_RejectedDTG = Now()
        'Else
            'Me.BC_RejectedDTG.Value = Me.BC_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "REJECTED"
        DoCmd.RunCommand acCmdSaveRecord
        Call rejectNoticeBCReq
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
                DoCmd.OpenForm "frm_RequirementOnly_Request_BC"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BC_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_RequirementOnly_Request_BC"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me.FunctionalAreaTo.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.FencedWBSTo.Requery
    Me.FencedWBSTo.Value = ""
    Me.FencedNameTo.Requery
    Me.FencedNameTo.Value = ""
    Me.WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me.ExtendedOrg.Requery
    Me.FunctionalAreaTo.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.FencedWBSTo.Requery
    Me.FencedWBSTo.Value = ""
    Me.FencedNameTo.Requery
    Me.FencedNameTo.Value = ""
End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "R (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database


Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()

    DoCmd.SetWarnings False
    Me.BC_Approved.Value = -1
    Me.BC_Name.Value = Environ("USERNAME")
    Me.BC_ApprovedDTG = Now()
    If IsNull(Me.BC_Comments) Then
        Me.BC_Comments.Value = "APPROVED"
    Else
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "APPROVED"
    End If
    DoCmd.RunCommand acCmdSaveRecord
    DoCmd.OutputTo acOutputReport, "rpt_RequirementOnlyBC_Form", acFormatPDF, "\\RUCKW0U9G67001\drm\PB
D\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\RequestBackups\" & Me.RID.Value & ".p
df"
    Call emailBCSubReq

    DoCmd.OpenQuery "qry_ReqOnlyIncreaseBC"
    Me.Form.Requery
    'DoCmd.GoToRecord , , acFirst

    addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
    If addRequests = vbYes Then
        DoCmd.OpenForm "frm_RequirementOnly_Request_BC_Sub"
        Me.Form.Requery

    Else
        DoCmd.Close acForm, "frm_RequirementOnly_Request_BC_Sub"
        DoCmd.OpenForm "frm_Transition"
    End If

    DoCmd.SetWarnings True

End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False
    If Me.BC_Approved = 0 And IsNull(Me.BC_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.BC_Approved = 0 And Not IsNull(Me.BC_Comments) Then
        Me.BC_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BC_RejectedDTG) Then
            Me.BC_RejectedDTG = Now()
        'Else
            'Me.BC_RejectedDTG.Value = Me.BC_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Approved.Value = 0
        Me.BC_Comments.Value = Me.BC_Comments.Value & vbCrLf & "REJECTED"
        DoCmd.RunCommand acCmdSaveRecord
        Call emailBCSubRejReq
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
```

```
                DoCmd.OpenForm "frm_RequirementOnly_Request_BC_Sub"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BC_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_RequirementOnly_Request_BC_Sub"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me.FunctionalAreaTo.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me.ExtendedOrg.Requery
    Me.FunctionalAreaTo.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.WBS.Requery
    Me.WBS.Value = ""

End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "R (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database


Private Sub Combo302_AfterUpdate()

    DoCmd.GoToRecord , , acLast

    If Me.Text348.Value <> Me.Combo302.Value Then

        Do

            DoCmd.GoToRecord , , acPrevious

        Loop Until Me.Text348.Value = Me.Combo302.Value

    End If

End Sub

Private Sub Command195_Click()

    DoCmd.SetWarnings False
    Me.BO_Approved.Value = -1
    Me.BO_Name.Value = Environ("USERNAME")
    Me.BO_ApprovedDTG = Now()
    If IsNull(Me.BO_Comments) Then
        Me.BO_Comments.Value = "APPROVED"
    Else
        Me.BO_Comments.Value = Me.BO_Comments.Value & vbCrLf & "APPROVED"
    End If
    DoCmd.RunCommand acCmdSaveRecord
    DoCmd.OutputTo acOutputReport, "rpt_RequirementOnlyBU_Form", acFormatPDF, "\\RUCKW0U9G67001\drm\PB
D\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\RequestBackups\" & Me.RID.Value & ".p
df"
    Call BOApprovalReq

    DoCmd.OpenQuery "qry_ReqOnlyIncrease"
    Me.Form.Requery
    'DoCmd.GoToRecord , , acFirst

    addRequests = MsgBox("Would you like to review additional requests?", vbYesNo)
    If addRequests = vbYes Then
        DoCmd.OpenForm "frm_RequirementOnly_Request_BO"
        Me.Form.Requery

    Else
        DoCmd.Close acForm, "frm_RequirementOnly_Request_BO"
        DoCmd.OpenForm "frm_Transition"
    End If

    DoCmd.SetWarnings True

End Sub

Private Sub Command213_Click()
DoCmd.SetWarnings False
    If Me.BO_Approved = 0 And IsNull(Me.BO_Comments) Then
        MsgBox "Please provide comments to support the rejection of this request.", vbOKOnly, "Rejecti
on Notice"
On Error GoTo errHandler:
    ElseIf Me.BO_Approved = 0 And Not IsNull(Me.BO_Comments) Then
        Me.BO_Name.Value = Environ("USERNAME")
        'If IsNull(Me.BO_RejectedDTG) Then
            Me.BO_RejectedDTG = Now()
        'Else
            'Me.BO_RejectedDTG.Value = Me.BO_RejectedDTG.Value & vbCrLf & Now()
        'End If
        Me.BC_Approved.Value = 0
        Me.BO_Comments.Value = Me.BO_Comments.Value & vbCrLf & "REJECTED"
        DoCmd.RunCommand acCmdSaveRecord
        Call rejectNoticeBOReq
        rejectMsg = MsgBox("Would you like to review additional requests?", vbYesNo)
            If rejectMsg = vbYes Then
```

```
                DoCmd.OpenForm "frm_RequirementOnly_Request_BO"
                'Me.Requery
                'DoCmd.GoToRecord , , acNext
            Else
                'Me.BO_Name.Value = Environ("USERNAME")
                'DoCmd.RunCommand acCmdSaveRecord
                'Me.Requery
                DoCmd.Close acForm, "frm_RequirementOnly_Request_BO"
                DoCmd.OpenForm "frm_Transition"
            End If
    Else
On Error GoTo errHandler:
        DoCmd.RunCommand acCmdSaveRecord
        'DoCmd.GoToRecord , , acNext
errHandler:
        Me.Requery
    End If
End Sub

Private Sub Command254_Click()
    DoCmd.Close
    DoCmd.OpenForm "frm_Transition"
End Sub

Private Sub ExtendedOrg_AfterUpdate()
    Me.FunctionalAreaTo.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.WBS.Requery
    Me.WBS.Value = ""
End Sub

Private Sub ParentOrg_AfterUpdate()
    Me.ExtendedOrg.Value = ""
    Me.ExtendedOrg.Requery
    Me.FunctionalAreaTo.Requery
    Me.FunctionalAreaTo.Value = ""
    Me.WBS.Requery
    Me.WBS.Value = ""

End Sub

Private Sub Command13_Click()
CreateObject("Shell.Application").Open ("{{0}}")

End Sub

Private Sub cmdAddAttachments_Click()
Dim oFS As FileSystemObject
Dim sPath As String

    sPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\R
equestAttachments\" & RID & "R (FY" & FYINFO(Now, "Y") & ")\"
    Set oFS = New FileSystemObject

    If oFS.FolderExists(sPath) Then
    Else
        Call oFS.CreateFolder(sPath)

    End If

    CreateObject("Shell.Application").Open (sPath)
End Sub
```

```
Option Compare Database


Private Sub Functional_Area_AfterUpdate()
    Me.SAG = Left([Functional Area], 3)
    Me.AMSCO = Left([Functional Area], 6)
    Me.MDEP = Right([Functional Area], 4)
    Me.Refresh
End Sub

Private Sub Object_Class_AfterUpdate()

    Select Case Me.Object_Class
        Case "11"
            Me.Object_Class_Description = "Civilian Pay"
        Case "21"
            Me.Object_Class_Description = "Travel"
        Case "22"
            Me.Object_Class_Description = "Transportation"
        Case "23"
            Me.Object_Class_Description = "Rental"
        Case "24"
            Me.Object_Class_Description = "Printing"
        Case "25"
            Me.Object_Class_Description = "Contracts"
        Case "26"
            Me.Object_Class_Description = "Supplies"
        Case "31"
            Me.Object_Class_Description = "Equipment"
        Case "Other"
            Me.Object_Class_Description = "Other"
        Case "WH"
            Me.Object_Class_Description = "Withhold"
    End Select
    Me.Refresh
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.OpenForm "frm_MainControl", acNormal
End Sub
```

```
Option Compare Database

Private Sub Command10_Click()
    DoCmd.OpenForm "frm_RequestDB_Main"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub


Private Sub Command128_Click()
    DoCmd.OpenForm "frm_ProgramOnly_Request_BC"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub

Private Sub Command157_Click()
    DoCmd.OpenForm "frm_RequirementOnly_Request_BC"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub
```

```
Option Compare Database

Private Sub Command10_Click()
    DoCmd.OpenForm "frm_RequestDB_Main"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub


Private Sub Command128_Click()
    DoCmd.OpenForm "frm_ProgramOnly_Request_BC"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub

Private Sub Command157_Click()
    DoCmd.OpenForm "frm_RequirementOnly_Request_BC"
    DoCmd.Close acForm, "frm_BC_UnapprovedCount"
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.Close , "frm_TableEdit_PnR"
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.Close , "frm_TableEdit_ProgramOnly"
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
    DoCmd.Close , "frm_TableEdit_RequirementOnly"
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Form_Close()
 DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Functional_Area_AfterUpdate()
        Me.SAG.Value = Left([Functional Area], 3)
        Me.AMSCO.Value = Left([Functional Area], 6)
        Me.MDEP.Value = Right([Functional Area], 4)
        Me.Refresh
End Sub
```

```
Option Compare Database

Private Sub Command0_Click()
    DoCmd.OpenForm "frm_MainControl"
    DoCmd.Close acForm, "frm_Transition"
End Sub

Private Sub Command3_Click()
    DoCmd.OpenForm "frm_BO_UnapprovedCount"
    DoCmd.Close acForm, "frm_Transition"
End Sub

Private Sub Command9_Click()
    DoCmd.OpenForm "frm_BCSub_UnapprovedCount"
    DoCmd.Close acForm, "frm_Transition"
End Sub
```

```
Option Compare Database

Private Sub Combo0_AfterUpdate()
'If Me.Combo0 = "" Then
    Select Case Me.Combo0
        Case "Program Only"
            DoCmd.OpenForm "frm_ProgramMonthly_Request", acNormal
        Case "Requirement Only"
            DoCmd.OpenForm "frm_RequirementOnly_Request", acNormal
        Case "Program and Requirement"
            DoCmd.OpenForm "frm_PgmAndReqMonthly_Request", acNormal
    End Select
    DoCmd.Close acForm, "frm_TypeRequest"
End Sub


Private Sub Command5_Click()
    DoCmd.OpenForm "frm_MainControl"
    DoCmd.Close acForm, "frm_TypeRequest"
End Sub
```

```
Option Compare Database
Option Compare Database

Private Sub Report_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database

Private Sub Report_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database
Option Compare Database
Option Compare Database
Option Compare Database

Private Sub Report_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Report_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Report_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Report_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Private Sub Report_Close()
    DoCmd.OpenForm "frm_MainControl"
End Sub
```

```
Option Compare Database

Public Function addPgmReqFolder()

    Dim strDir As String
    strDir = "C:\ProgramRequests\"
    If Dir(strDir, vbDirectory) = "" Then MkDir strDir

End Function
```

```
Option Compare Database

'------------------------------------------------------------
' AttachmentPurge - removes attachments from records over a specified number of days old.
'               This function performs the following actions:
'                       1. queries each of the three program and requirements (PnR)tables for records
over a specified number of days old (without the attachments)
'                       2. appends the above records to the "tbl_ArchiveRequests" table
'                       3. deletes the records from step 1 from their original/respective tables in or
der to remove the attachments
'                       4. appends the remaining records from each of the three PnR tables to the "tbl
_ALL_Requests" table
'------------------------------------------------------------
Function AttachmentPurge()
On Error GoTo AttachmentPurge_Err

DoCmd.SetWarnings False

    Call execQuery("01a_qry_ProgramOnly_OverAge_Append")
    Call execQuery("01b_qry_ReqOnly_OverAge_Append")
    Call execQuery("01c_qry_PnR_OverAge_Append")
    Call execQuery("01a_qry_ProgramOnly_OverAge_Delete")
    Call execQuery("01b_qry_ReqOnly_OverAge_Delete")
    Call execQuery("01c_qry_PnR_OverAge_Delete")
    Call execQuery("000_qry_All_Requests_Delete")
    Call execQuery("001_qry_Pgm_UpdateALL")
    Call execQuery("002_qry_Req_UpdateALL")
    Call execQuery("003_qry_PnR_UpdateALL")
    MsgBox "Program and Requirement Requests over 30 days old have been purged.", vbOKOnly, "Update No
tice"

DoCmd.SetWarnings True
AttachmentPurge_Exit:
    Exit Function

AttachmentPurge_Err:
    MsgBox Error$
    Resume AttachmentPurge_Exit

End Function
```

```
Option Compare Database

Private Const dbLog As Boolean = True 'Set this to False to turn all logging off.
Public Function backendBackupSOF()
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String              'To calculate processing time
Dim dbPath As String                'To capture the current application path
Dim dbName As String                'To capture the current application name
Dim engPath As String               'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()   'record the start time
    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\2_Engines\SOF_Database_BE\SOF_Database_BE.accdb", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPE
RATIONS BRANCH\00 PBD SOF Databse\RTS Database\7_Backup_Engines\SOF_Database_BE_Backup" & (Format(Now,
 "yyyymmdd_hhmm")) & ".accdb")
    endTime = Now() 'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss") 'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "backendBackupSOF"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If
End Function
Public Function BEBackupNewSOF()
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String              'To calculate processing time
Dim dbPath As String                'To capture the current application path
Dim dbName As String                'To capture the current application name
Dim engPath As String               'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()  'record the start time
    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\2_Engines\SOF_Database_BE\SOF_Database_BE.accdb", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPE
RATIONS BRANCH\00 PBD SOF Databse\RTS Database\7_Backup_Engines\SOF_Engine" & (Format(Now, "yyyymmdd_h
hmm")) & ".accdb")
    endTime = Now()  'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss")  'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "backendBackupSOF"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If
End Function
Public Function cleanBackups()
Dim FileSys As FileSystemObject
Dim objFile As file
Dim myFolder
Dim strFileName As String
Dim dteFile As Date
Dim myDir As String
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String              'To calculate processing time
Dim dbPath As String                'To capture the current application path
Dim dbName As String                'To capture the current application name
Dim engPath As String               'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName

    startTime = Now()  'record the start time
     'set path for file backups
    myDir = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\7
_Backup_Engines\"

    'set up filesys objects
    Set FileSys = New FileSystemObject
    Set myFolder = FileSys.GetFolder(myDir)


    'loop through each file and get date last modified. If largest date then store Filename
    dteFile = DateAdd("d", -5, Date)
    For Each objFile In myFolder.Files
        If objFile.DateLastModified < dteFile Then
            strFileName = objFile.Name
            'Debug.Print strFilename
            Kill (myFolder & "\" & strFileName)
        End If
    Next objFile

    Set FileSys = Nothing
    Set myFolder = Nothing

    endTime = Now()  'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss")  'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "cleanBackups"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

End Function
Public Function backendPRRBackup()
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String              'To calculate processing time
Dim dbPath As String                'To capture the current application path
Dim dbName As String                'To capture the current application name
Dim engPath As String               'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()  'record the start time
    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\2_Engines\PRR_Engine\PRR_BE\PRR_BE.accdb", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS
 BRANCH\00 PBD SOF Databse\RTS Database\7_Backup_Engines\PRR_BE_Backup" & (Format(Now, "yyyymmdd_hhmm"
)) & ".accdb")
    endTime = Now()  'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss")  'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "backendPRRBackup"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If
End Function

Function BackupTime() ' To capture and log the current time when the PRR Backup was conducted.  Used t
o display the time on the Main Control form.
Dim currDB As DAO.Database
Dim rs As Recordset
Set currDB = CurrentDb
Dim SQL As String
Dim tSQL As String

DoCmd.SetWarnings False

tSQL = "INSERT INTO tbl_BackupLog (LastBackupTime)" & "VALUES (Now())"
    DoCmd.RunSQL tSQL
SQL = ("SELECT * FROM qry_PRR_Last_BU")
    Set rs = CurrentDb.OpenRecordset(SQL)
    With rs
        If Not LastBackupTime = "" Then
            DoCmd.Quit
        End If
    End With
    Forms![frm_MainControl].Requery

    DoCmd.SetWarnings True
End Function
```

```
Option Compare Database

Function checkVersion(ByVal strFile As String)

    'StrFile = dir("\\RUCKW0U9G67001\drm\PBD SOF Database\SOF_Database_Frontend\" & "\*")
    strFile = Dir("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\4_ControlEngines\RTS_v*")
    checkVersion = strFile
    'Do While Len(StrFile) > 0
        'Debug.Print StrFile
    'Loop
    'MsgBox StrFile, vbOKOnly
End Function

Function currDB(ByVal DB_Name As String) 'retrieves the current DB name

    DB_Name = Application.CurrentProject.Name
    currDB = DB_Name

End Function

Function DBAcheck() As Boolean
Dim rs As Recordset
DBAcheck = True

'This is to make debugging by looking at previous versions easier. Allows the Database Admin to turn o
ff the version check easily
SQL = ("SELECT * FROM qry_EnvironAdminCheck")
Set rs = CurrentDb.OpenRecordset(SQL)
With rs
If !Position = "Contractor" Then
DBAcheck = answer = MsgBox("Run Version Check?", vbYesNo, "Version Control")
End If
End With

End Function


Function versionMsg() 'This function is used to determine if the user is utilizing the latest DB versi
on, if not, redirects to download the latest version.
Dim currDB As DAO.Database
Dim rs As Recordset
Set currDB = CurrentDb
Dim SQL As String

If DBAcheck Then
SQL = ("SELECT * FROM qry_VersionValue")
    Set rs = CurrentDb.OpenRecordset(SQL)
    With rs
        If !Version_Value >= 1 Then
            MsgBox "There is newer DB version available. You will be redirectd to download the latest
version. Copy this new version onto your desktop.", vbOKOnly
            Call Open_Explorer
            DoCmd.Quit
        End If
    End With
End If
End Function

Sub Open_Explorer() 'opens windows explorer to the specified path to download the latest version of th
e USAACE RTS DB (Control Engine)
    Dim PID As Double
    Dim strRootPath As String

    Const strExpExe = "explorer.exe"
    Const strArg = " " '" /e,/root, "

    '// Change rootpath here
    strRootPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\4_ControlEngines\"

    PID = Shell(strExpExe & strArg & strRootPath, 3)
```

```
modCheckVersion - 2

End Sub
```

```
Option Compare Database
Public Function FYINFO(ByVal d As Variant, t As String) As Integer 'converts the current date to the s
pecified Fiscal month or year.
    Select Case t
        Case "P": FYINFO = (Month(d) + 2) Mod 12 + 1
        Case "Y": FYINFO = (Year(d) + IIf(Month(d) > 9, 1, 0)) Mod 100
        Case "M": FYINFO = MonthName(Month(d), True)
    End Select
End Function

Public Function PriorFY(ByVal d As Variant, t As String) As Integer 'converts the current date to the
specified Fiscal month or year minus 1.
    Select Case t
        Case "P": PriorFY = (Month(d) + 2) Mod 12 + 1
        Case "Y": PriorFY = (Year(d) + IIf(Month(d) > 9, 1, 0)) Mod 100 - 1
        Case "M": PriorFY = MonthName(Month(d), True)
    End Select
End Function
```

```
Option Compare Database
Option Explicit

Const strURL As String = "https://army.deps.mil/Army/cmds/arsouth_g8/pbd/beb/SharedDocuments/Pivot_Tab
les/"

Public Sub Move_To_Sharepoint(ByVal strLocal As String, ByVal strFileName As String)
    Dim sharepointFileName
    Dim LlFileLength As Long
    Dim Lvarbin() As Byte
    Dim LobjXML As Object
    Dim LvarBinData As Variant
    Dim PstrFullfileName As String
    Dim PstrTargetURL As String
    Dim Password As String
    Dim Username As String

    Set LobjXML = CreateObject("Microsoft.XMLHTTP")
    sharepointFileName = strURL & strFileName

    PstrFullfileName = strLocal & strFileName
    LlFileLength = FileLen(PstrFullfileName) - 1

    ' Read the file into a byte array.
    ReDim Lvarbin(LlFileLength)
    Open PstrFullfileName For Binary As #1
    Get #1, , Lvarbin
    Close #1

    ' Convert to variant to PUT.
    LvarBinData = Lvarbin
    PstrTargetURL = strURL & strFileName

    ' Put the data to the server, false means synchronous.
    LobjXML.Open "PUT", PstrTargetURL, False, Username, Password

    ' Send the file in.
    LobjXML.Send LvarBinData

err_Copy:
     If Err <> 0 Then
       MsgBox Err & " " & Err.Description
     End If
End Sub
```

```
Option Compare Database

Public Function emailDBUpdate()
'*****************************************************************************************
'
' -This function generates a SOF Update email notification to recipients based on a
'  Yes/No requirement to receive the notice in the "Daily" column of the "tbl_Emails".
'
'*****************************************************************************************

Dim dailyMail As MailItem
Dim strEmail As String
Dim strBody As String
Dim strSubject As String
Dim rs As Recordset
Dim rs1 As Recordset
Dim grt As String

processNotice = MsgBox("Would you like to send a Database Update Notification?", vbYesNo, "Process Not
ice")
If processNotice = vbYes Then
    Call updateStatusBar("Generating email.....")
    Set dailyMail = Outlook.Application.CreateItem(olMailItem)
    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Move through Email table to gather emails to poplulate distribtion of the email.
        With rs
            If Not .EOF And Not .BOF Then
                .MoveLast
                .MoveFirst
                    For i = 0 To .RecordCount - 1
                        If !Daily = True Then
                          strEmail = strEmail & !Email & ";"
                          .MoveNext
                        Else: .MoveNext
                        End If
                    Next
            End If
        End With

    'Retrieve the text to use in the body of the email based on the type selected from a dropdown
    SQL1 = "SELECT * From tbl_EmailMessage"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    'With grt
        If Time <= TimeValue("11:59:59 AM") Then
            grt = "Good Morning,"
        Else
            grt = "Good Afternoon,"
        End If
    'End With
    'Combine email body and subject based on message
    With rs1
        strSubject = "**SOF DATABASE UPDATE NOTICE**"
    End With
    'Change the following email information as needed
          With dailyMail
               .Subject = strSubject
               '(Format(Now, "dd MMMM yy"))
               .To = strEmail
               '.Attachments.Add ("\\RUCKW0U9G67001\drm\PBD\DATA CALL\PAE\Daily Reports\SAG Recap\SAG
" & (Format(Now, "mm-dd-yyyy"))) & ".xlsx"
               .Importance = olImportanceHigh
               '.Send
               .Display
               'Sets body of email equal to message type selected with daily numbers and signature blo
ck of sender
               .HTMLBody = grt & "<b> <br><br><br>The SOF database has been updated as of: " & " " & T
ime & _
               "</b><br><br>Refresh for the most current data.<br><br><br>v/r," & "<br><br>Financial O
perations"

          End With
    ElseIf vbNo Then
    Call clearStatusBar
    End If
End Function
```

```
Option Compare Database

Public Sub emailConsumption()
'*********************************************************************************************
'
' -This function generates an email notification to recipients based on a Yes/No requirement
'  to receive the Daily Consumption email in the "Consumption" column of the "tbl_Emails".
'
'*********************************************************************************************
Dim dailyMail As MailItem
Dim strEmail As String
Dim strBody As String
Dim strSubject As String
Dim rs As Recordset
Dim rs1 As Recordset
Dim fs As New FileSystemObject
Dim strPath As String

strPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\Exports & Report
s\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(Now, "P") & (Format(Now, "mm
myyyy")) & "\"

    Set dailyMail = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Move through Email table to gather emails to poplulate distribtion of the email.
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Consumption = True Then
                      strEmail = strEmail & !Email & ";"
                      .MoveNext
                    ElseIf !Consumption = False Then
                        .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type selected from a dropdown
    SQL1 = "SELECT * From tbl_Temp"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)

    greeting = getGreeting
    'Combine email body and subject based on message
    With rs1
        strBody = greeting & !MsgBody
        strSubject = !MsgSubject
    End With

    'Change the following email information as needed
          With dailyMail
                    .Subject = strSubject & (Format(Now, "dd MMMM yy"))
                    .To = strEmail
                    .Importance = olImportanceHigh
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody

                 'IF statement to verify whether the Aviation file exists prior to attempting attachme
nt to the email
                    If fs.FileExists(strPath & "Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf") The
n
                        .Attachments.Add strPath & "\Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    Else
                    End If

                    If fs.FileExists(strPath & "128th " & (Format(Now, "mm-dd-yyyy")) & ".pdf") Then
                        .Attachments.Add strPath & "\128th " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    Else
                    End If

                    If fs.FileExists(strPath & "AERO " & (Format(Now, "mm-dd-yyyy")) & ".pdf") Then
                        .Attachments.Add strPath & "\AERO " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    Else
                    End If

                    If fs.FileExists(strPath & "COE " & (Format(Now, "mm-dd-yyyy")) & ".pdf") Then
                        .Attachments.Add strPath & "\COE " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    Else
                    End If

                    If fs.FileExists(strPath & "GCSS Army A57VV - " & (Format(Now, "yyyymmdd")) & ".pd
f") Then
                        .Attachments.Add strPath & "\GCSS Army A57VV - " & (Format(Now, "yyyymmdd")) &
 ".pdf"
                    Else
                    End If

                    If fs.FileExists(strPath & "GCSS Army A57VB - " & (Format(Now, "yyyymmdd")) & ".pd
f") Then
                        .Attachments.Add strPath & "\GCSS Army A57VB - " & (Format(Now, "yyyymmdd")) &
 ".pdf"
                    Else
                    End If

          End With
End Sub


Public Sub emailConsumptionOld()

'Not used anymore.  Updated to above code.
'*********************************************************************************************
'
' -This function generates an email notification to recipients based on a Yes/No requirement
'  to receive the Daily Consumption email in the "Consumption" column of the "tbl_Emails".
'
'*********************************************************************************************
Dim dailyMail As MailItem
Dim strEmail As String
Dim strBody As String
Dim strSubject As String
Dim rs As Recordset
Dim rs1 As Recordset
Dim fs As New FileSystemObject

    Set dailyMail = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Move through Email table to gather emails to poplulate distribtion of the email.
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Consumption = True Then
                      strEmail = strEmail & !Email & ";"
                      .MoveNext
                    ElseIf !Consumption = False Then
                        .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type selected from a dropdown
    SQL1 = "SELECT * From tbl_Temp"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)

    'Combine email body and subject based on message
    With rs1
        strBody = !MsgBody
        strSubject = !MsgSubject
    End With

    'Change the following email information as needed
          With dailyMail
                'IF statement to verify whether the Aviation file exists prior to attempting attachmen
t to the email
                If fs.FileExists("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF D
atabse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(Now,
"P") & _
                        (Format(Now, "mmmyyyy")) & "\" & "Aviation " & (Format(Now, "mm-dd-yyyy")) & "
.pdf") Then
                    .Subject = strSubject
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\128th " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\AERO " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\COE " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\GCSS Army A57VB - " & (Format(Now, "yyyymmdd")) &
 ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\GCSS Army A57VV - " & (Format(Now, "yyyymmdd")) &
 ".pdf"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody
               Else
                    .Subject = strSubject & " " & (Format(Now, "dd MMMM yy"))
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\128th " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\AERO " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\COE " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\GCSS Army A57VB - " & (Format(Now, "yyyymmdd")) &
 ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\GCSS Army A57VV - " & (Format(Now, "yyyymmdd")) &
 ".pdf"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody
               End If
          End With
End Sub
```

```
Option Compare Database

Public Sub emailBCSubPRMon()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program and Requirement Request has been approved/submitted by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BC_S
ub].[AnalystName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReqMonthly_Request_BC_Sub].[RID].Value & " was APPROVED by " & [Forms]![frm_PgmAndReqMonthly_Request
_BC_Sub].[BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
         'Change the following email information as needed

    With Email
              .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqMonthly_Request_BC_Sub].[Reason]
& "***"
              .To = strEmail
              '.Attachments.Add ""
              .Importance = olImportanceHigh
              '.Send
              .Display
              .Body = strBody
              If Not strFromEmail = "" Then Set .SentOnBehalfOf = strFromEmail
    End With
End Sub

Public Sub emailBCSubRejPRMon()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program and Requirement Request has been rejected by the Branch Chief (BC)
'
'*********************************************************************************************
```

```
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BC_Sub].[Analys
tName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReqMonthly_Request_BC_Sub].[RID].Value & " was REJECTED by " & [Forms]![frm_PgmAndReqMonthly_Request
_BC_Sub].[BC_Name].Value & "."
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqMonthly_Request_BC_Sub].[Re
ason] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub emailBCSubReq()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program and Requirement Request has been approved/submitted by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_RequirementOnly_Request_BC_Su
b].[AnalystName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Requ
irementOnly_Request_BC_Sub].[RID].Value & " was APPROVED by " & [Forms]![frm_RequirementOnly_Request_B
C_Sub].[BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_RequirementOnly_Request_BC_Sub].[Rea
son] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub emailBCSubRejReq()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Requirement Only Request has been rejected by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_RequirementOnly_Request_BC_Sub].[Analyst
Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Requ
irementOnly_Request_BC_Sub].[RID].Value & " was REJECTED by " & [Forms]![frm_RequirementOnly_Request_B
C_Sub].[BC_Name].Value & "."
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_RequirementOnly_Request_BC_Sub].[Rea
son] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub emailBCSubPgm()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program Only Request has been approved by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_ProgramMonthly_Request_BC_Sub
].[AnalystName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramMonthly_Request_BC_Sub].[RID].Value & " was APPROVED by " & [Forms]![frm_ProgramMonthly_Request_BC_
Sub].[BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramMonthly_Request_BC_Sub].[Reas
on] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub emailBCSubRejPgm()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program Only Request has been rejected by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_ProgramMonthly_Request_BC_Sub].[AnalystN
ame].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramMonthly_Request_BC_Sub].[RID].Value & " was REJECTED by " & [Forms]![frm_ProgramMonthly_Request_BC_
Sub].[BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
```

```
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramMonthly_Request_BC_Sub].[Reas
on] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub
```

```
Option Compare Database

Public Sub BCApprovalPgmAndReq()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program and Requirement Request has been approved by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !BC_Approval = True Or !Username = [Forms]![frm_PgmAndReq_Request_BC].[AnalystN
ame].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'BC_Approval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReq_Request_BC].[RID].Value & " was APPROVED by " & [Forms]![frm_PgmAndReq_Request_BC].[BC_Name].Val
ue & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReq_Request_BC].[Reason] & "**
*"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub requestNoticePgmAndReq()
'*********************************************************************************************
'
' -This function generates an "Request Notice" email notification to recipients when a
'  request from the PRR DB has been submitted by an Analyst.
'
'*********************************************************************************************

 Dim Email As MailItem
```

```
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !AwaitingApproval = True Or !BC_Approval = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'AwaitingApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - " & [Forms]![frm_PgmAndReq_Request]
.[AnalystName].Value & " has submitted Request ID:  #" & [Forms]![frm_PgmAndReq_Request].[RID].Value &
 " for approval." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReq_Request].[Reason].Value &
"***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBCPgmAndReq()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program and Requirement Request has been rejected by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_PgmAndReq_Request_BC].[AnalystName].Valu
e Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReq_Request_BC].[RID].Value & " was REJECTED by " & [Forms]![frm_PgmAndReq_Request_BC].[BC_Name].Val
ue & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReq_Request_BC].[Reason] & "**
*"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBOPgmAndReq()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program and Requirement Request has been rejected by the Budget Officer (BO)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_PgmAndReq_Request_BO].[AnalystName].Valu
e Or !Username = [Forms]![frm_PgmAndReq_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReq_Request_BO].[RID].Value & " was REJECTED by " & [Forms]![frm_PgmAndReq_Request_BO].[BO_Name].Val
ue & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReq_Request_BO].[Reason] & "**
*"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub BOApprovalPgmAndReq()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program and Requirement Request has been approved by the Budget Officer (BO)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_PgmAndReq_Request_BO].[Analys
tName].Value Or !Username = [Forms]![frm_PgmAndReq_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value
    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReq_Request_BO].[RID].Value & " was APPROVED by " & [Forms]![frm_PgmAndReq_Request_BO].[BO_Name].Val
ue & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReq_Request_BO].[Reason] & "**
*"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub
```

```
Option Compare Database

Public Sub BCApproval()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program Only Request has been approved by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !BC_Approval = True Or !Username = [Forms]![frm_ProgramOnly_Request_BC].[Analys
tName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'BC_Approval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramOnly_Request_BC].[RID].Value & " was APPROVED by " & [Forms]![frm_ProgramOnly_Request_BC].[BC_Name]
.Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramOnly_Request_BC].[Reason] & "
***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub requestNotice()
'*********************************************************************************************
'
' -This function generates an "Request Notice" email notification to recipients when a
'  request from the PRR DB has been submitted by an Analyst.
'
'*********************************************************************************************
 Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !AwaitingApproval = True Or !BC_Approval = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'AwaitingApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - " & [Forms]![frm_ProgramOnly_Reques
t].[AnalystName].Value & " has submitted Request ID:  #" & [Forms]![frm_ProgramOnly_Request].[RID].Val
ue & " for approval." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramOnly_Request].[Reason].Value
& "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBC()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program Only Request has been rejected by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
```

```
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_ProgramOnly_Request_BC].[AnalystName].Va
lue Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramOnly_Request_BC].[RID].Value & " was REJECTED by " & [Forms]![frm_ProgramOnly_Request_BC].[BC_Name]
.Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramOnly_Request_BC].[Reason] & "
***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBO()
'**********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program Only Request has been rejected by the Budget Officer (BO)
'
'**********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_ProgramOnly_Request_BO].[AnalystName].Va
lue Or !Username = [Forms]![frm_ProgramOnly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramOnly_Request_BO].[RID].Value & " was REJECTED by " & [Forms]![frm_ProgramOnly_Request_BO].[BO_Name]
.Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramOnly_Request_BO].[Reason] & "
***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub BOApproval()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program Only Request has been approved by the Budget Officer (BO)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_ProgramOnly_Request_BO].[Anal
ystName].Value Or !Username = [Forms]![frm_ProgramOnly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value
    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramOnly_Request_BO].[RID].Value & " was APPROVED by " & [Forms]![frm_ProgramOnly_Request_BO].[BO_Name]
.Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramOnly_Request_BO].[Reason] & "
***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub
```

```
Option Compare Database

Public Sub BCApprovalPgmMon()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program Only Request has been approved by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !BC_Approval = True Or !Username = [Forms]![frm_ProgramMonthly_Request_BC].[Ana
lystName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'BC_Approval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramMonthly_Request_BC].[RID].Value & " was APPROVED by " & [Forms]![frm_ProgramMonthly_Request_BC].[BC
_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramMonthly_Request_BC].[Reason]
& "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub requestNoticePgmMon()
'*********************************************************************************************
'
' -This function generates an "Request Notice" email notification to recipients when a
'  request from the PRR DB has been submitted by an Analyst.
'
'*********************************************************************************************
 Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !AwaitingApproval = True Or !BC_Approval = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'AwaitingApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - " & [Forms]![frm_ProgramMonthly_Req
uest].[AnalystName].Value & " has submitted Request ID:  #" & [Forms]![frm_ProgramMonthly_Request].[RI
D].Value & " for approval." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramMonthly_Request].[Reason].Val
ue & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBCPgmMon()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program Only Request has been rejected by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
```

```
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_ProgramMonthly_Request_BC].[AnalystName]
.Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramMonthly_Request_BC].[RID].Value & " was REJECTED by " & [Forms]![frm_ProgramMonthly_Request_BC].[BC
_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramMonthly_Request_BC].[Reason]
& "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBOPgmMon()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program Only Request has been rejected by the Budget Officer (BO)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_ProgramMonthly_Request_BO].[AnalystName]
.Value Or !Username = [Forms]![frm_ProgramMonthly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramMonthly_Request_BO].[RID].Value & " was REJECTED by " & [Forms]![frm_ProgramMonthly_Request_BO].[BO
_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramMonthly_Request_BO].[Reason]
& "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub BOApprovalPgmMon()
'*******************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program Only Request has been approved by the Budget Officer (BO)
'
'*******************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_ProgramMonthly_Request_BO].[A
nalystName].Value Or !Username = [Forms]![frm_ProgramMonthly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value
    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Prog
ramMonthly_Request_BO].[RID].Value & " was APPROVED by " & [Forms]![frm_ProgramMonthly_Request_BO].[BO
_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_ProgramMonthly_Request_BO].[Reason]
& "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub
```

```
Option Compare Database

Public Sub BCApprovalPgmReqMon()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program Only Request has been approved by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !BC_Approval = True Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BC].[A
nalystName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'BC_Approval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReqMonthly_Request_BC].[RID].Value & " was APPROVED by " & [Forms]![frm_PgmAndReqMonthly_Request_BC]
.[BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqMonthly_Request_BC].[Reason
] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub requestNoticePgmReqMon()
'*********************************************************************************************
'
' -This function generates an "Request Notice" email notification to recipients when a
'  request from the PRR DB has been submitted by an Analyst.
'
'*********************************************************************************************
 Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !AwaitingApproval = True Or !BC_Approval = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'AwaitingApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - " & [Forms]![frm_PgmAndReqMonthly_R
equest].[AnalystName].Value & " has submitted Request ID:  #" & [Forms]![frm_PgmAndReqMonthly_Request]
.[RID].Value & " for approval." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqMonthly_Request].[Reason].V
alue & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub requestNoticePgmReqFMT()
'*********************************************************************************************
'
' -This function generates an "Request Notice" email notification to recipients when a
'  request from the PRR DB has been submitted by an Analyst.
'
'*********************************************************************************************
 Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
```

```
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !AwaitingApproval = True Or !BC_Approval = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'AwaitingApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - " & [Forms]![frm_PgmAndReqFMT_Reque
st].[AnalystName].Value & " has submitted Request ID:  #" & [Forms]![frm_PgmAndReqFMT_Request].[RID].V
alue & " for approval." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqFMT_Request].[Reason].Value
 & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBCPgmReqMon()
'******************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program Only Request has been rejected by the Branch Chief (BC)
'
'******************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BC].[AnalystNam
e].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReqMonthly_Request_BC].[RID].Value & " was REJECTED by " & [Forms]![frm_PgmAndReqMonthly_Request_BC]
.[BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqMonthly_Request_BC].[Reason
] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBOPgmReqMon()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Program Only Request has been rejected by the Budget Officer (BO)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BO].[AnalystNam
e].Value Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
```

```
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReqMonthly_Request_BO].[RID].Value & " was REJECTED by " & [Forms]![frm_PgmAndReqMonthly_Request_BO]
.[BO_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqMonthly_Request_BO].[Reason
] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub BOApprovalPgmReqMon()
'*********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Program Only Request has been approved by the Budget Officer (BO)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BO].
[AnalystName].Value Or !Username = [Forms]![frm_PgmAndReqMonthly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_PgmA
ndReqMonthly_Request_BO].[RID].Value & " was APPROVED by " & [Forms]![frm_PgmAndReqMonthly_Request_BO]
.[BO_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_PgmAndReqMonthly_Request_BO].[Reason
] & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub
```

```
Option Compare Database

Public Sub BCApprovalReq()
'**********************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Requirement Only Request has been approved by the Branch Chief (BC)
'
'**********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !BC_Approval = True Or !Username = [Forms]![frm_RequirementOnly_Request_BC].[An
alystName].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'BC_Approval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Requ
irementOnly_Request_BC].[RID].Value & " was APPROVED by " & [Forms]![frm_RequirementOnly_Request_BC].[
BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_RequirementOnly_Request_BC].[Reason]
 & "***"
                    .To = strEmail
                    '.Attachments.Add "R"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub requestNoticeReq()
'**********************************************************************************************
'
' -This function generates an "Request Notice" email notification to recipients when a
'  request from the PRR DB has been submitted by an Analyst.
'
'**********************************************************************************************
 Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !AwaitingApproval = True Or !BC_Approval = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'AwaitingApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - " & [Forms]![frm_RequirementOnly_Re
quest].[AnalystName].Value & " has submitted Request ID:  #" & [Forms]![frm_RequirementOnly_Request].[
RID].Value & " for approval." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_RequirementOnly_Request].[Reason].Va
lue & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBCReq()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Requirement Only Request has been rejected by the Branch Chief (BC)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
```

```
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_RequirementOnly_Request_BC].[AnalystName
].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Requ
irementOnly_Request_BC].[RID].Value & " was REJECTED by " & [Forms]![frm_RequirementOnly_Request_BC].[
BC_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_RequirementOnly_Request_BC].[Reason]
 & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub rejectNoticeBOReq()
'*********************************************************************************************
'
' -This function generates an "Rejected" email notification to recipients when a
'  Requirement Only Request has been rejected by the Budget Officer (BO)
'
'*********************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Rejection Or !Username = [Forms]![frm_RequirementOnly_Request_BO].[AnalystName
].Value Or !Username = [Forms]![frm_RequirementOnly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'Rejection'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value

    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Requ
irementOnly_Request_BO].[RID].Value & " was REJECTED by " & [Forms]![frm_RequirementOnly_Request_BO].[
BO_Name].Value & "."  '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_RequirementOnly_Request_BO].[Reason]
 & "***"
                    .To = strEmail
                    '.Attachments.Add ""
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub

Public Sub BOApprovalReq()
'*******************************************************************************************
'
' -This function generates an "Approval" email notification to recipients when a
'  Requirement Only Request has been approved by the Budget Officer (BO)
'
'*******************************************************************************************
Dim Email As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject

    Set Email = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Gathers email recipient's email addresses to poplulate distribtion of the email.
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !FinalApproval = True Or !Username = [Forms]![frm_RequirementOnly_Request_BO].[
AnalystName].Value Or !Username = [Forms]![frm_RequirementOnly_Request_BO].[BC_Name].Value Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type
    SQL1 = "SELECT * From tbl_OutlookBody WHERE Type = 'FinalApproval'"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim requestID As Integer
    Dim aName As String
    'requestID = Me!requestID.Value
    'aName = Me!AnalystName.Value
    With rs1
            strBody = !MsgBody & vbCrLf & vbCrLf & "Request Info - Request ID:  #" & [Forms]![frm_Requ
irementOnly_Request_BO].[RID].Value & " was APPROVED by " & [Forms]![frm_RequirementOnly_Request_BO].[
BO_Name].Value & "." '& vbCrLF *****add request details here*****
            strSubject = !MsgSubject
    End With
          'Change the following email information as needed
          With Email
                    .Subject = strSubject & " ***" & [Forms]![frm_RequirementOnly_Request_BO].[Reason]
 & "***"
                    .To = strEmail
                    '.Attachments.Add "R"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    .Body = strBody
          End With
End Sub
```

```
Option Compare Database

Public Sub emailSAGRecap()
'*********************************************************************************************
'
' -This function generates an email notification to recipients based on a Yes/No requirement
'  to receive the SAG Recap email in the "SAGRecap" column of the "tbl_Emails".
'
'*********************************************************************************************
Dim dailyMail As Object
Dim strEmail As String
Dim strBody As String
Dim strSubject As String
Dim rs As Recordset
Dim rs1 As Recordset


    Set dailyMail = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Move through Email table to gather emails to poplulate distribtion of the email.
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !SAGRecap = True Then
                      strEmail = strEmail & !Email & ";"
                      .MoveNext
                    ElseIf !Consumption = False Then
                        .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type selected from a dropdown
    SQL1 = "SELECT * From tbl_OutlookBody"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)

    'Combine email body and subject based on message
    With rs1
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst

            For i = 0 To .RecordCount - 1
                If !Type = "SAG Recap" Then
                    strBody = strBody & !MsgBody
                    strSubject = !MsgSubject
                    i = .RecordCount - 1
                Else
                    .MoveNext
                End If
            Next
         End If

    End With

    greeting = getGreeting
    'Change the following email information as needed
          With dailyMail
               .Subject = strSubject & " " & (Format(Now, "dd MMMM yy"))
               .To = strEmail
               .Attachments.Add ("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF D
atabse\Exports & Reports\Daily Reports\SAG Recap\SAG " & (Format(Now, "mm-dd-yyyy"))) & ".xlsx"
               .Importance = olImportanceHigh
               '.Send
               .Display
               'Sets body of email equal to message type selected with daily numbers and signature blo
ck of sender
               .Body = greeting & strBody
          End With
End Sub

Public Sub emailManDBUpdate()
'**********************************************************************************************
'
' -This function generates an email notification to recipients based on a Yes/No requirement
'  to receive the Daily email in the "Daily" column of the "tbl_Emails".
'
'**********************************************************************************************
Dim dailyMail As Object
Dim strEmail As String
Dim strBody As String
Dim strSubject As String
Dim rs As Recordset
Dim rs1 As Recordset
Dim fs As New FileSystemObject

    Set dailyMail = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Move through Email table to gather emails to poplulate distribtion of the email.
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Daily = True Then
                      strEmail = strEmail & !Email & ";"
                      .MoveNext
                    ElseIf !Consumption = False Then
                        .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type selected from a dropdown
    SQL1 = "SELECT * From tbl_OutlookBody"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)

    strBody = getGreeting


    'Combine email body and subject based on message
    With rs1
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst

            For i = 0 To .RecordCount - 1
                If !Type = "Database Update" Then
                    strBody = strBody & !MsgBody
                    strSubject = !MsgSubject
                    i = .RecordCount - 1
                Else
                    .MoveNext
                End If
            Next
         End If

    End With

    strPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\Exports & Re
ports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(Now, "P") & (Format(Now,
 "mmmyyyy")) & "\"

    'Change the following email information as needed
    With dailyMail
         .Subject = strSubject
         .To = strEmail
         '.Send
         .Display
         'Sets body of email equal to message type selected with daily numbers and signature block of 
sender
         .Body = strBody

         'IF statement to verify whether the Aviation file exists prior to attempting attachment to th
e email
            If fs.FileExists(strPath & "Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf") Then
                .Attachments.Add strPath & "\Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
            Else
            End If

    End With
End Sub
```

```
Option Compare Database

Public Sub emailSOFOld() 'This has been replaced by the function below
    Dim dailyMail As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject
    Dim quote As String

    Set dailyMail = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Move through Email table to gather emails to poplulate distribtion of the email.
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Daily = True Or !EOMCleanup = True Or !EOMFinalized = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type selected from a dropdown
    SQL1 = "SELECT * From tbl_Temp"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim cVal As String
    'Set wb = Workbooks.Open("\\RUCKW0U9G67001\drm\PBD\DATA CALL\PAE\Daily Reports\Morning Numbers\FY"
 & FYINFO(Now, "Y") & "\" & FYINFO(Now, "P") & (Format(Now, "mmmyyyy")) & _
            "\Morning Numbers " & (Format(Now, "yyyymmdd")) & ".xlsx")
    Set wb = Workbooks.Open("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databs
e\Exports & Reports\Daily Reports\Morning Numbers\Morning Numbers " & (Format(Now, "yyyymmdd")) & ".xl
sx")
    Worksheets("sheet1").Range("C45:G46").Copy
    cVal = Range("C45").Value
    cVal1 = Range("C46").Value
    cVal2 = Range("E45").Value
    cVal3 = Range("E46").Value
    cVal4 = Range("G45").Value
    cVal5 = Range("G46").Value

    'Retrieve daily quote from Daily Quotes table in current database based on numeric day of the mont
h
    SQL2 = "SELECT * FROM tbl_DailyQuotes"
    Set rs2 = CurrentDb.OpenRecordset(SQL2)
    With rs2
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !ID = DatePart("d", Date) Then
                        quote = !Quotes
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With

    'Combine email body based on message selection type with additional daily data from the current Mo
rning Numbers sheet
    With rs1
        'If !Type = "Daily" Then
            strBody = !MsgBody & vbCrLf & "Today's allotment and obligations totals: " _
                & vbCrLf & vbCrLf & "Allotment" & vbTab & vbTab & Format(cVal, "Currency") _
                & vbCrLf & "Obligation" & vbTab & vbTab & Format(cVal1, "Currency") _
                & vbCrLf & vbCrLf & "Yesterday's allotment and obligations totals: " _
                & vbCrLf & vbCrLf & "Allotment" & vbTab & vbTab & Format(cVal2, "Currency") _
                & vbCrLf & "Obligation" & vbTab & vbTab & Format(cVal3, "Currency") _
                & vbCrLf & vbCrLf & "Difference: " _
                & vbCrLf & vbCrLf & "Allotment" & vbTab & vbTab & Format(cVal4, "Currency") _
                & vbCrLf & "Obligation" & vbTab & vbTab & Format(cVal5, "Currency") _
                & vbCrLf & vbCrLf & vbCrLf & vbCrLf & "Thought of the day:  " & quote & vbCrLf & vbCrL
f
        'Else: strBody = !MsgBody
        'End If
            strSubject = !MsgSubject

    End With
    'Saves Morning Numbers file without saving. Only opened and imported numbers, no changes; no need
to save
    With wb
        .Saved = True
        .Close
    End With

          'Change the following email information as needed
          With dailyMail
                'IF statement to verify whether the Aviation file exists prior to attempting attachmen
t to the email
                If fs.FileExists("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF D
atabse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(Now,
"P") & _
                        (Format(Now, "mmmyyyy")) & "\" & "Aviation " & (Format(Now, "mm-dd-yyyy")) & "
.pdf") Then
                    .Subject = strSubject
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Negative Blotter\Negative " & (Format(Now, "mm-dd-yyyy")) &
 ".xlsx"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody
               Else
                    .Subject = strSubject
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Negative Blotter\Negative " & (Format(Now, "mm-dd-yyyy")) &
 ".xlsx"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody
               End If
          End With
End Sub


Public Sub emailSOF()
'*****************************************************************************************************
'
' -This function generates an email notification to recipients based on a Yes/No requirement
'  to receive the email updates in the "EOMCleanup" or "EOMFinalized" column of the "tbl_Emails".
'
'*****************************************************************************************************
    Dim dailyMail As MailItem
    Dim strEmail As String
    Dim strBody As String
    Dim strSubject As String
    Dim rs As Recordset
    Dim rs1 As Recordset
    Dim fs As New FileSystemObject
    Dim quote As String
    Dim avn As String
    Dim od As String
    Dim file As Integer


    Set dailyMail = Outlook.Application.CreateItem(olMailItem)

    SQL = "SELECT * FROM tbl_Emails"
    Set rs = CurrentDb.OpenRecordset(SQL)
    'Move through Email table to gather emails to poplulate distribtion of the email.
    With rs
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
                    If !Daily = True Or !EOMCleanup = True Or !EOMFinalized = True Then
                        strEmail = strEmail & !Email & ";"
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    'Retrieve the text to use in the body of the email based on the type selected from a dropdown
    SQL1 = "SELECT * From tbl_Temp"
    Set rs1 = CurrentDb.OpenRecordset(SQL1)
    Dim wb As Workbook
    Dim cVal As String
    'Set wb = Workbooks.Open("\\RUCKW0U9G67001\drm\PBD\DATA CALL\PAE\Daily Reports\Morning Numbers\FY"
 & FYINFO(Now, "Y") & "\" & FYINFO(Now, "P") & (Format(Now, "mmmyyyy")) & _
            "\Morning Numbers " & (Format(Now, "yyyymmdd")) & ".xlsx")
    Set wb = Workbooks.Open("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databs
e\Exports & Reports\Daily Reports\Morning Numbers\Morning Numbers " & (Format(Now, "yyyymmdd")) & ".xl
sx")
    Worksheets("sheet1").Range("C36:G37").Copy
    cVal = Range("C36").Value
    cVal1 = Range("C37").Value
    cVal2 = Range("E36").Value
    cVal3 = Range("E37").Value
    cVal4 = Range("G36").Value
    cVal5 = Range("G37").Value

    'Retrieve daily quote from Daily Quotes table in current database based on last used quote
    DoCmd.SetWarnings False
    'figures out what the last quote used was and uses the next one or cycles back up to the first quo
te
    SQLrotate = ("SELECT * FROM qry_LastQuote as lastQuote")
    grabbing = CurrentDb.OpenRecordset(SQLrotate)
    putting = grabbing("[LastUsed]").Value

    maxQuotes = DCount("*", "tbl_DailyQuotes")

    If putting = maxQuotes Then
    newQuote = 1
    Else
    newQuote = putting + 1
    End If

    tSQL = "INSERT INTO tbl_DailyQuotesTrack (Quote_Number, Date_Used)" & "VALUES (" & newQuote & ", N
ow())"
    DoCmd.RunSQL tSQL

    SQL2 = "SELECT * FROM tbl_DailyQuotes"
    Set rs2 = CurrentDb.OpenRecordset(SQL2)
    With rs2
        If Not .EOF And Not .BOF Then
            .MoveLast
            .MoveFirst
                For i = 0 To .RecordCount - 1
```

```
                    If !ID = newQuote Then
                        quote = !Quotes
                        .MoveNext
                    Else: .MoveNext
                    End If
                Next
        End If
    End With
    DoCmd.SetWarnings True

    greeting = getGreeting
    'Combine email body based on message selection type with additional daily data from the current Mo
rning Numbers sheet
    With rs1
        'If !Type = "Daily" Then
            strBody = greeting & !MsgBody & vbCrLf & "Today's allotment and obligations totals: " _
                & vbCrLf & vbCrLf & "Allotment" & vbTab & vbTab & Format(cVal, "Currency") _
                & vbCrLf & "Obligation" & vbTab & vbTab & Format(cVal1, "Currency") _
                & vbCrLf & vbCrLf & "Yesterday's allotment and obligations totals: " _
                & vbCrLf & vbCrLf & "Allotment" & vbTab & vbTab & Format(cVal2, "Currency") _
                & vbCrLf & "Obligation" & vbTab & vbTab & Format(cVal3, "Currency") _
                & vbCrLf & vbCrLf & "Difference: " _
                & vbCrLf & vbCrLf & "Allotment" & vbTab & vbTab & Format(cVal4, "Currency") _
                & vbCrLf & "Obligation" & vbTab & vbTab & Format(cVal5, "Currency") _
                & vbCrLf & vbCrLf & vbCrLf & vbCrLf & "Thought of the day:  " & quote & vbCrLf & vbCrL
f
        'Else: strBody = !MsgBody
        'End If
            strSubject = !MsgSubject

    End With
    'Saves Morning Numbers file without saving. Only opened and imported numbers, no changes; no need
to save

    'Dim strMonthName As String

    'If Month(Now) = 1 Then

        'strMonthName = MonthName(12)

   ' Else

       ' strMonthName = MonthName(Month(Now) - 1)

    'End If

   ' MsgBox "Reminder for EoM emails, replace the month in the email with the correct month: " & strMo
nthName & "."

    With wb
        .Saved = True
        .Close
    End With
    avn = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\Exports & Report
s\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(Now, "P") & _
                        (Format(Now, "mmmyyyy")) & "\" & "Aviation " & (Format(Now, "mm-dd-yyyy")) & "
.pdf"
    od = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\Exports & Reports
\Daily Reports\Outstanding Request\Outstanding Requests " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
          'Change the following email information as needed
          With dailyMail
                'IF statement to verify whether the Aviation file exists prior to attempting attachmen
t to the email
                If fs.FileExists(avn) And fs.FileExists(od) Then
                    file = 2
                ElseIf fs.FileExists(avn) Then
                    file = 1
                ElseIf fs.FileExists(od) Then
                    file = 3
                Else
                    file = 0
                End If
                If file = 2 Then
                    .Subject = strSubject
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Outstanding Request\Outstanding Requests " & (Format(Now, "
mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Negative Blotter\Negative " & (Format(Now, "mm-dd-yyyy")) &
 ".xlsx"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody

                ElseIf file = 1 Then
                    .Subject = strSubject
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Consumption Reports\FY" & FYINFO(Now, "Y") & "\" & FYINFO(N
ow, "P") & _
                        (Format(Now, "mmmyyyy")) & "\Aviation " & (Format(Now, "mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Negative Blotter\Negative " & (Format(Now, "mm-dd-yyyy")) &
 ".xlsx"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody
                ElseIf file = 3 Then
                    .Subject = strSubject
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Outstanding Request\Outstanding Requests " & (Format(Now, "
mm-dd-yyyy")) & ".pdf"
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Negative Blotter\Negative " & (Format(Now, "mm-dd-yyyy")) &
 ".xlsx"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody
               Else
                    .Subject = strSubject
                    .To = strEmail
                    .Attachments.Add "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD S
OF Databse\Exports & Reports\Daily Reports\Negative Blotter\Negative " & (Format(Now, "mm-dd-yyyy")) &
 ".xlsx"
                    .Importance = olImportanceHigh
                    '.Send
                    .Display
                    'Sets body of email equal to message type selected with daily numbers and signatur
e block of sender
                    .Body = strBody
               End If
          End With

End Sub

Function getGreeting()
If Time <= TimeValue("11:59:59 AM") Then
            getGreeting = "Good Morning"
        Else
            getGreeting = "Good Afternoon"
```

```
    End If
End Function
```

```
Option Compare Database

Public Function getLogon() 'retrieves the Username of the logged on user
Dim uName As String
Dim pcName As String
    getLogon = Environ("USERNAME")
    'pcName = Environ("COMPUTERNAME")
    'Debug.Print getLogon
    'Debug.Print pcName
End Function

Public Function adminCheck() 'verifies whether a user has Admin rights, then displays the appropriate
pages on the Main Control form
Dim currDB As DAO.Database
Dim rs As Recordset
Set currDB = CurrentDb
Dim SQL As String

SQL = ("SELECT * FROM qry_EnvironAdminCheck")
    Set rs = CurrentDb.OpenRecordset(SQL)
    With rs
        If !Admin = -1 Then
             Forms("frm_MainControl").Admin.Visible = True
             Forms("frm_MainControl").SOF_Interface.Visible = True
             Forms("frm_MainControl").[SOF Forms/Reports].Visible = True
             Forms("frm_MainControl").FMT_Data.Visible = True
             Forms("frm_MainControl").[PRR Selection/Status].Visible = True
             Forms("frm_MainControl").[PRR Forms/Reports].Visible = True
        Else
             Forms("frm_MainControl").[PRR Selection/Status].Visible = True
             Forms("frm_MainControl").[PRR Forms/Reports].Visible = True
             Forms("frm_MainControl").Admin.Visible = False
             Forms("frm_MainControl").SOF_Interface.Visible = False
             Forms("frm_MainControl").FMT_Data.Visible = False
             Forms("frm_MainControl").[SOF Forms/Reports].Visible = False
        End If
    End With
End Function
```

```
Option Compare Database

Public Function deleteExistingTbls()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\SOF_Engine\SOF_Engine.acc
db", False
'Run Sub procedure.
    app.Run "deleteTables"
Set app = Nothing
End Function
```

```
Option Compare Database

Private Const dbLog As Boolean = True 'Set this to False to turn all logging off.
'------------------------------------------------------------
' keyTableExport
'
'------------------------------------------------------------
Function keyTableExport()

On Error GoTo keyTableExport_Err
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "Cost Collector Information", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G6
7001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\Cost
 Collector Information.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "FMT - Obligation Log - Fenced", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U
9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\F
MT - Obligation Log - Fenced.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "FMT - Obligation Log - Negative Fenced", "ExcelWorkbook(*.xlsx)", "
\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update
Reports\FMT - Obligation Log - Negative Fenced.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "Functional Area Information", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G
67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\Fun
ctional Area Information.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "Parent Extended and Branch Info", "ExcelWorkbook(*.xlsx)", "\\RUCKW
0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports
\Parent Extended and Branch Info.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "TIGER Program Details", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G67001\
drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\TIGER Pro
gram Details.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "tbl_Requirements_and_Program", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9
G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\Re
quirements & Program.xlsx", False, "", , acExportQualityPrint

    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "keyTableExport"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

keyTableExport_Exit:
    Exit Function

keyTableExport_Err:
    MsgBox Error$
    Resume keyTableExport_Exit

End Function

Function exportCC()

On Error GoTo exportCC_Err
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "02_tbl_CostCollectorInformation", "ExcelWorkbook(*.xlsx)", "\\RUCKW
0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports
\Cost Collector Information.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportCC"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

exportCC_Exit:
    Exit Function

exportCC_Err:
    MsgBox Error$
    Resume exportCC_Exit

End Function

Function exportCI()

On Error GoTo exportCI_Err
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "Committment Item Information", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9
G67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & FYINFO(Now, "Y") & " Database Files\Committment Item
Information.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportCI"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If
```

```
exportCI_Exit:
    Exit Function

exportCI_Err:
    MsgBox Error$
    Resume exportCI_Exit

End Function

Function exportFMTlog()

On Error GoTo exportFMTlog_Err
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "FMT - Obligation Log - Fenced", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U
9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\F
MT - Obligation Log - Fenced.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "FMT - Obligation Log - Negative Fenced", "ExcelWorkbook(*.xlsx)", "
\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update
Reports\FMT - Obligation Log - Negative Fenced.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportFMTlog"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

exportFMTlog_Exit:
    Exit Function

exportFMTlog_Err:
    MsgBox Error$
    Resume exportFMTlog_Exit

End Function

Function exportFA()

On Error GoTo exportFA_Err
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "Functional Area Information", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G
67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\Fun
ctional Area Information.xlsx", False, "", , acExportQualityPrint
```

```
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportFA"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

exportFA_Exit:
    Exit Function

exportFA_Err:
    MsgBox Error$
    Resume exportFA_Exit

End Function

Function exportParentOrg()

On Error GoTo exportParentOrg_Err
Dim startTime As Date                   'To record processing time
Dim endTime As Date                     'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "Parent Extended and Branch Info", "ExcelWorkbook(*.xlsx)", "\\RUCKW
0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports
\Parent Extended and Branch Info.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportParentOrg"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

exportParentOrg_Exit:
    Exit Function

exportParentOrg_Err:
    MsgBox Error$
    Resume exportParentOrg_Exit

End Function


Function exportPrgmDetails()

On Error GoTo exportPrgmDetails_Err
Dim startTime As Date                   'To record processing time
Dim endTime As Date                  'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "TIGER Program Details", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G67001\
drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\TIGER Pro
gram Details.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportPrgmDetails"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

exportPrgmDetails_Exit:
    Exit Function

exportPrgmDetails_Err:
    MsgBox Error$
    Resume exportPrgmDetails_Exit

End Function

Function exportPhaseObReport()

On Error GoTo exportPhaseObReport_Err
Dim startTime As Date                'To record processing time
Dim endTime As Date                  'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "13_PhaseObReport", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G67001\drm\P
BD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\TIGER Phase Ob
ligations Report.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportPhaseObReport"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If
```

```
exportPhaseObReport_Exit:
    Exit Function

exportPhaseObReport_Err:
    MsgBox Error$
    Resume exportPhaseObReport_Exit

End Function

Function exportPhaseCommReport()

On Error GoTo exportPhaseCommReport_Err
Dim startTime As Date                'To record processing time
Dim endTime As Date                  'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "14_PhaseCommReport", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G67001\drm
\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\TIGER Phase
Commitments Report.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportPhaseCommReport"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

exportPhaseCommReport_Exit:
    Exit Function

exportPhaseCommReport_Err:
    MsgBox Error$
    Resume exportPhaseCommReport_Exit

End Function

Function exportPnR()

On Error GoTo exportPnR_Err
Dim startTime As Date                'To record processing time
Dim endTime As Date                  'To record processing time
Dim execTime As String
Dim dbPath As String
Dim dbName As String
Dim engPath As String

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()
    DoCmd.SetWarnings False
    DoCmd.OutputTo acOutputTable, "tbl_Requirements_and_Program", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9
G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\Re
quirements & Program.xlsx", False, "", , acExportQualityPrint
    endTime = Now()
    execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportPnR"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

exportPnR_Exit:
    Exit Function

exportPnR_Err:
    MsgBox Error$
    Resume exportPnR_Exit

End Function
```

```
Option Compare Database

Public Function importFHP()
On Error GoTo importFHP_Err
SetWarnings = False

    DoCmd.TransferSpreadsheet acImport, 10, "tbl_FHP_Temp", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPE
RATIONS BRANCH\00 PBD SOF Databse\Exports & Reports\Morning Numbers Tools\SOF - Morning Validation.xls
x", False, "Negative Flying Hour Prg!C:M"
importFHP_Exit:
    Exit Function

importFHP_Err:
    MsgBox Error$
    Resume importFHP_Exit

End Function
```

```
Option Compare Database

Public Function newFHPParts()

        Dim sFile As String
        Dim sFolder As String
        Dim dFile As String
        Dim dFolder As String

        'Define source/destiniation folder and file names
        sFile = "FHP Parts Form Template.xlsm"
        sFolder = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\1_Data\FHP - Parts Form\Template\"
        dFile = "FHP Parts Form " & Format(Now(), "yyyyMMdd hh-mm-ss") & ".xlsm"
        dFolder = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\1_Data\FHP - Parts Form\"

        'Create Object for File System
        Set fso = CreateObject("Scripting.FileSystemObject")

        'Checking If File Is Located in the Source Folder
        If Not fso.FileExists(sFolder & sFile) Then
            MsgBox "Specified File Not Found in Source Folder", vbInformation, "Not Found"
            End

        'Copying If the Same File is Not Located in the Destination Folder
        ElseIf Not fso.FileExists(dFolder & dFile) Then
            fso.CopyFile (sFolder & sFile), (dFolder & dFile), True

            Dim appExcel As Excel.Application
            Dim myWorkbook As Excel.Workbook

            Set appExcel = CreateObject("Excel.Application")
            Set myWorkbook = appExcel.Workbooks.Open(dFolder & dFile)
            appExcel.Visible = True
            Set appExcel = Nothing
            Set myWorkbook = Nothing
        Else
            MsgBox "Specified File Already Exists In The Destination Folder", vbExclamation, "File Alr
eady Exists"
        End If

End Function

Public Function newCivPayForm()

        Dim sFile As String
        Dim sFolder As String
        Dim dFile As String
        Dim dFolder As String

        'Define source/destiniation folder and file names
        sFile = "CivPay Form Template.xlsm"
        sFolder = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\1_Data\CivPay Form\Template\"
        dFile = "CivPay Form " & Format(Now(), "yyyyMMdd hh-mm-ss") & ".xlsm"
        dFolder = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\1_Data\CivPay Form\"

        'Create Object for File System
        Set fso = CreateObject("Scripting.FileSystemObject")

        'Checking If File Is Located in the Source Folder
        If Not fso.FileExists(sFolder & sFile) Then
            MsgBox "Specified File Not Found in Source Folder", vbInformation, "Not Found"
            End

        'Copying If the Same File is Not Located in the Destination Folder
        ElseIf Not fso.FileExists(dFolder & dFile) Then
            fso.CopyFile (sFolder & sFile), (dFolder & dFile), True

            Dim appExcel As Excel.Application
            Dim myWorkbook As Excel.Workbook
            Set appExcel = CreateObject("Excel.Application")
            Set myWorkbook = appExcel.Workbooks.Open(dFolder & dFile)
            appExcel.Visible = True
            Set appExcel = Nothing
            Set myWorkbook = Nothing
        Else
            MsgBox "Specified File Already Exists In The Destination Folder", vbExclamation, "File Alr
eady Exists"
        End If

End Function

Public Function submitMultiRequestForm(fileLoc As String)

    Dim fd As Office.FileDialog

    Set fd = Application.FileDialog(msoFileDialogFilePicker)

    Call updateStatusBar("File selection.....")
    With fd

        .AllowMultiSelect = False

        .InitialFileName = fileLoc

        ' Set the title of the dialog box.
        .Title = "Please select the file."

        ' Show the dialog box. If the .Show method returns True, the
        ' user picked at least one file. If the .Show method returns
        ' False, the user clicked Cancel.
        If .Show = True Then
            sFile = .SelectedItems(1)

        Else

            End

        End If

    End With

    Dim oExcel As Object
    Dim oExcelWrkBk As Object
    Dim oExcelWrkSht As Object
    Dim oExcelShtSig As Object
    Dim oExcelShtPgm As Object
    Dim oExcelShtReq As Object
    Dim oExcelShtPnR As Object
    Dim bExcelOpened As Boolean

    'Start Excel
    On Error Resume Next
    Set oExcel = GetObject(, "Excel.Application")    'Bind to existing instance of Excel
    If Err.Number <> 0 Then    'Could not get instance of Excel, so create a new one
        Err.Clear
        Set oExcel = CreateObject("Excel.Application")
    Else
        bExcelOpened = True    'GetObject worked -> Excel was already running
    End If
    On Error GoTo Error_Handler

    oExcel.Visible = False   'Keep Excel hidden until we are done with our manipulation
    oExcel.ScreenUpdating = False

    Call updateStatusBar("Pulling request information.....")
    Set oExcelWrkBk = oExcel.Workbooks.Open(sFile)    'Open the Workbook
    Set oExcelShtSig = oExcelWrkBk.Sheets("Signatures")    'Get the right WorkSheet to work with
    Set oExcelShtPgm = oExcelWrkBk.Sheets("Program")    'Get the right WorkSheet to work with
    Set oExcelShtReq = oExcelWrkBk.Sheets("Requirement")    'Get the right WorkSheet to work with
    Set oExcelShtPnR = oExcelWrkBk.Sheets("Pgm&Req")    'Get the right WorkSheet to work with
    Dim exBA, exBADate, exBC, exBCDate, exBCComm, exBO, exBODate, exBOComm, exReqCount As String

    Dim people(1 To 3, 1 To 3) As String

    'Names
    people(1, 1) = oExcelShtSig.Range("C7").Value       'Budget Analyst
    people(2, 1) = oExcelShtSig.Range("C8").Value       'Branch Chief
    people(3, 1) = oExcelShtSig.Range("C9").Value       'Budget Officer

    'Dates
    people(1, 2) = oExcelShtSig.Range("D7").Value   'Budget Analyst
    people(2, 2) = oExcelShtSig.Range("D8").Value   'Branch Chief
    people(3, 2) = oExcelShtSig.Range("D9").Value   'Budget Officer

    'Comments (typically just "APPROVED")
    people(2, 3) = Replace(Replace(oExcelShtSig.Range("E8").Value, """", ""), "'", "")  'Branch Chief
    people(3, 3) = Replace(Replace(oExcelShtSig.Range("E9").Value, """", ""), "'", "")  'Budget Office
r

    If people(2, 1) = "" Or people(2, 2) = "" Or people(2, 3) = "" Then

        MsgBox ("This form has not yet been approved by the BC. Either have the BC complete the form o
r select a different form to submit.")

        DoCmd.SetWarnings True

        oExcelWrkBk.Close False    'Close the WorkBook without saving now that we're done.

        If bExcelOpened = False Then    'Close excel if is wasn't originally running
               oExcel.Quit
        End If

        oExcel.Visible = True
        oExcel.ScreenUpdating = True

        Call clearStatusBar

        Exit Function

    End If

    'Check to make sure all data is filled out

    If oExcelShtPgm.Range("I8").Value > 0 Then

        Call missingCheck(oExcelShtPgm, bExcelOpened, oExcel)

    End If

    If oExcelShtReq.Range("I8").Value > 0 Then

        Call missingCheck(oExcelShtReq, bExcelOpened, oExcel)

    End If

    If oExcelShtPnR.Range("I8").Value > 0 Then

        Call missingCheck(oExcelShtPnR, bExcelOpened, oExcel)

    End If

    'Start importing requests

    If oExcelShtPgm.Range("I8").Value > 0 Then

        Call multiRequestInput(oExcelWrkBk, oExcelShtPgm, people)

    End If

    If oExcelShtReq.Range("I8").Value > 0 Then

        Call multiRequestInput(oExcelWrkBk, oExcelShtReq, people)
    End If

    If oExcelShtPnR.Range("I8").Value > 0 Then

        Call multiRequestInput(oExcelWrkBk, oExcelShtPnR, people)

    End If

    DoCmd.SetWarnings True

    oExcelWrkBk.Close False    'Close the WorkBook without saving now that we're done.

    If bExcelOpened = False Then    'Close excel if is wasn't originally running
        oExcel.Quit
    End If

    oExcel.Visible = True
    oExcel.ScreenUpdating = True

    Call updateStatusBar("Creating archive.....")
    Dim fso As Object
    Set fso = CreateObject("Scripting.Filesystemobject")

    fileBackStop = InStrRev(sFile, "\")
    sFileName = Right(sFile, Len(sFile) - fileBackStop)
    sFolderName = fileLoc

    fso.MoveFile Source:=sFolderName & sFileName, Destination:=sFolderName & "Archive\" & sFileName

    Call clearStatusBar
    Exit Function

Error_Handler_Exit:
    On Error Resume Next
    If Not oExcelWrkSht Is Nothing Then Set oExcelWrkSht = Nothing
    If Not oExcelWrkBk Is Nothing Then Set oExcelWrkBk = Nothing
    If Not oExcel Is Nothing Then
        oExcel.ScreenUpdating = True
        oExcel.Visible = True    'Make excel visible to the user
        Set oExcel = Nothing
    End If
    Exit Function

Error_Handler:
    '9       -> can't find the worksheet
    '1004    -> can't find the file
    If Err.Number = 9 Then
       MsgBox "The following error has occurred" & vbCrLf & vbCrLf & _
               "Error Number: " & Err.Number & vbCrLf & _
               "Error Source: Excel_GetRangeVal" & vbCrLf & _
               "Error Description: Unable to locate the specified WorkSheet '" & sSht & "'" & _
               Switch(Erl = 0, "", Erl <> 0, vbCrLf & "Line No: " & Erl) _
               , vbOKOnly + vbCritical, "An Error has Occurred!"
        GoTo Error_Handler_Exit
    Else
          MsgBox "The following error has occurred" & vbCrLf & vbCrLf & _
               "Error Number: " & Err.Number & vbCrLf & _
               "Error Source: Excel_GetRangeVal" & vbCrLf & _
               "Error Description: " & Err.Description & _
               Switch(Erl = 0, "", Erl <> 0, vbCrLf & "Line No: " & Erl) _
               , vbOKOnly + vbCritical, "An Error has Occurred!"
        Resume Error_Handler_Exit
    End If

End Function

Public Function AddMinute(ByVal sTime As String) As String
    Dim dt As Date

    dt = CDate(sTime)
    dt = DateAdd("n", 1, dt)

    AddMinute = Format(dt, "mm/dd/yy h:nnam/pm")
```

```
End Function

Sub missingCheck(oExcelWrkSht As Object, bExcelOpened As Boolean, oExcel As Object)

    exReqCount = oExcelWrkSht.Range("I8").Value 'Gets the number of requests for this sheet

    lRow = oExcelWrkSht.Cells(100000, 1).End(xlUp).row 'Gets the last row of the table

    'If there is only one record it will do something else later

    If lRow = 12 Then

        lRow = 13
        oneRecord = True

    End If

    Dim requestArr() As Variant
    requestArr = oExcelWrkSht.Range("A12:S" & lRow).Value 'Copies the table into an array

    DoCmd.SetWarnings False

    Dim ToFrom, FAToFrom, strSQL, RID As String

    For i = 1 To lRow - 11

        For j = 1 To 19

            If requestArr(i, j) = "" Or IsNull(requestArr(i, j)) Then

                colName = " (""" & oExcelWrkSht.Cells(11, j).Value & """)"

                MsgBox ("Column " & j & colName & " Row " & i & " of sheet " & oExcelWrkSht.Name & " i
s missing data. Have the analyst complete the form and try again.")

                'Go through the normal routine to close Excel if some of the data is missing
                DoCmd.SetWarnings True
                If bExcelOpened = False Then oExcel.Quit
                oExcel.Visible = True
                oExcel.ScreenUpdating = True

                End

            End If

        Next j

        If oneRecord Then i = i + 1

    Next i

End Sub

Sub multiRequestInput(oExcelWrkBk As Object, oExcelWrkSht As Object, ByRef peopleInt() As String)

    exReqCount = oExcelWrkSht.Range("I8").Value 'Gets the number of requests for this sheet

    lRow = oExcelWrkSht.Cells(100000, 1).End(xlUp).row 'Gets the last row of the table

    'If there is only one record it will do something else later

    If lRow = 12 Then

        lRow = 13
        oneRecord = True

    End If

    Dim requestArr() As Variant
    requestArr = oExcelWrkSht.Range("A12:S" & lRow).Value 'Copies the table into an array

    DoCmd.SetWarnings False
    Dim ToFrom, FAToFrom, strSQL, RID As String

    For i = 1 To lRow - 11

        Call updateStatusBar("Uploading " & oExcelWrkSht.Name & " request " & i & " of " & exReqCount
& ".....")

        For j = 1 To 19

            requestArr(i, j) = Replace(Replace(requestArr(i, j), """", ""), "'", "''") 'Replaces any d
ouble tick quotes to single tick quotes for SQL purposes

        Next j

        Dim inputTable As String
        Dim pgmCheck As Boolean

        Select Case oExcelWrkSht.Name 'Adds variables based on the sheet being input

            Case "Program"
                inputTable = "tbl_ProgramMonthly_Input"
                typeLabel = "P"
                labelLen = 4
                pgmCheck = True
                reqCheck = False

            Case "Requirement"
                inputTable = "tbl_RequirementOnly_Input"
                typeLabel = "R"
                labelLen = 4
                pgmCheck = False
                reqCheck = True

            Case "Pgm&Req"
                inputTable = "tbl_PgmAndReqMonthly_Input"
                typeLabel = "PR"
                labelLen = 5
                pgmCheck = True
                reqCheck = True

        End Select

        'Gets the RID based on the last request input in the appropriate table
        strRID = "SELECT MAX(CINT(LEFT(" & inputTable & ".RID, LEN(" & inputTable & ".RID)-" & labelLe
n & "))) As maxRID FROM " & inputTable
        Set rst = CurrentDb.OpenRecordset(strRID)
        RID = rst!maxRID + 1
        rst.Close

        'Gets the fiscal year for the RID
        If Month(Date) < 10 Then
            strFY = Right(Year(Date), 2)
        Else
            strFY = Right(Year(Date) + 1, 2)
        End If

        RID = RID & "-" & strFY & typeLabel


        'Input Table
        If pgmCheck Then

            strSQL = "INSERT INTO " & inputTable & " " _
            & "(RID, FundCenter, ParentOrg, ExtendedOrg, Fund, Reason, WBS, FundingSource, [AMO Packet
], MissionCriticality, FundingCategory" _
            & ", FunctionalArea, ObjectClass, FencedName, FencedWBS, " & requestArr(i, 2) & "Pgm, Rema
rks" _
            & ", BC_Comments, BO_Comments, AnalystName, BC_Name, BO_Name, BC_Approved, BO_Approved, Re
questDTG, BC_ApprovedDTG, BO_ApprovedDTG)" _
            & " VALUES ('" & RID & "', '" & requestArr(i, 3) & "', '" & requestArr(i, 4) & "', '" & re
questArr(i, 5) & "', '" & requestArr(i, 6) & "', '" & requestArr(i, 7) _
            & "', '" & requestArr(i, 14) & "', '" & requestArr(i, 8) & "', 'No', '" & requestArr(i, 9)
```

```
 & "', '" & requestArr(i, 10) & "', '" & requestArr(i, 15) _
            & "', '" & requestArr(i, 11) & " - " & requestArr(i, 12) & "', '" & requestArr(i, 13) & "'
, '" & requestArr(i, 14) & "', '" & requestArr(i, 17) & "', '" & requestArr(i, 19) _
            & "', '" & peopleInt(2, 3) & "', '" & peopleInt(3, 3) & "', '" & peopleInt(1, 1) & "', '"
& peopleInt(2, 1) & "', '" & peopleInt(3, 1) _
            & "', TRUE, TRUE, '" & peopleInt(1, 2) & "', '" & peopleInt(2, 2) & "', '" & peopleInt(3,
2) & "')"

        Else

            strSQL = "INSERT INTO " & inputTable & " " _
            & "(RID, FundCenter, ParentOrg, ExtendedOrg, Fund, Reason, WBS, FundingSource, [AMO Packet
], MissionCriticality, FundingCategory" _
            & ", FunctionalAreaTo, ObjectClassTo, FencedNameTo, FencedWBSTo, AmountTo, TotalChangeTo,
Remarks" _
            & ", BC_Comments, BO_Comments, AnalystName, BC_Name, BO_Name, BC_Approved, BO_Approved, Re
questDTG, BC_ApprovedDTG, BO_ApprovedDTG)" _
            & " VALUES ('" & RID & "', '" & requestArr(i, 3) & "', '" & requestArr(i, 4) & "', '" & re
questArr(i, 5) & "', '" & requestArr(i, 6) & "', '" & requestArr(i, 7) _
            & "', '" & requestArr(i, 14) & "', '" & requestArr(i, 8) & "', 'No', '" & requestArr(i, 9)
 & "', '" & requestArr(i, 10) & "', '" & requestArr(i, 15) _
            & "', '" & requestArr(i, 11) & " - " & requestArr(i, 12) & "', '" & requestArr(i, 13) & "'
, '" & requestArr(i, 14) & "', '" & requestArr(i, 16) _
            & "', '" & requestArr(i, 17) & "', '" & requestArr(i, 19) _
            & "', '" & peopleInt(2, 3) & "', '" & peopleInt(3, 3) & "', '" & peopleInt(1, 1) & "', '"
& peopleInt(2, 1) & "', '" & peopleInt(3, 1) _
            & "', TRUE, TRUE, '" & peopleInt(1, 2) & "', '" & peopleInt(2, 2) & "', '" & peopleInt(3,
2) & "')"

        End If

        DoCmd.RunSQL strSQL

        If pgmCheck Then

            'tbl_Requirements_and_Program
            strSQL = "INSERT INTO tbl_Requirements_and_Program " _
            & "([Fund Center], Fund, [Funding Source], [Parent Organization], [Extended Organization],
 [Cost Collector], SAG, [Functional Area], AMSCO, MDEP, [Object Class], [Object Class Description]" _
            & ", [Funding Category], Fenced, [Detailed Program], [Entered By], [Fiscal Month], [Progra
m Update Quarter], [Date], Notes, [Mission Criticality])" _
            & " VALUES ('" & requestArr(i, 3) & "', '" & requestArr(i, 6) & "', '" & requestArr(i, 8)
& "', '" & requestArr(i, 4) & "', '" & requestArr(i, 5) & "', '" & requestArr(i, 14) & "', '" _
            & Left(requestArr(i, 15), 3) & "', '" & requestArr(i, 15) & "', '" & Left(requestArr(i, 15
), 6) & "', '" & Right(requestArr(i, 15), 4) & "', '" & requestArr(i, 11) & "', '" _
            & requestArr(i, 12) & "', '" & requestArr(i, 10) & "', '" & requestArr(i, 13) & "', '" & r
equestArr(i, 17) & "', '" & RID & Left(requestArr(i, 2), 3) & "', '" & requestArr(i, 2) & "', '" _
            & requestArr(i, 7) & "', '" & peopleInt(3, 2) & "', '" & requestArr(i, 19) & "', '" & requ
estArr(i, 9) & "')"


            peopleInt(2, 2) = AddMinute(peopleInt(2, 2))
            'peopleInt(3, 2) = AddMinute(peopleInt(3, 2)) For BO who was taken out of the approval pro
cess. Keeping in case they bring them back.

            DoCmd.RunSQL strSQL

            'Withhold - tbl_Requirements_and_Program
            strSQL = "INSERT INTO tbl_Requirements_and_Program " _
            & "([Fund Center], Fund, [Funding Source], [Parent Organization], [Extended Organization],
 SAG, [Functional Area], AMSCO, MDEP, [Object Class], [Object Class Description]" _
            & ", [Funding Category], Fenced, [Detailed Program], [Entered By], [Fiscal Month], [Progra
m Update Quarter], [Date], Notes, [Mission Criticality])" _
            & " VALUES ('" & requestArr(i, 3) & "', '" & requestArr(i, 6) & "', '" & requestArr(i, 8)
& "', 'G8 Withhold', 'G8 Withhold', '" _
            & Left(requestArr(i, 15), 3) & "', '" & requestArr(i, 15) & "', '" & Left(requestArr(i, 15
), 6) & "', '" & Right(requestArr(i, 15), 4) & "', 'WH', 'Withhold', '" _
            & requestArr(i, 10) & "', 'Unfenced', '" & requestArr(i, 17) * -1 & "', '" & RID & "W" & L
eft(requestArr(i, 2), 3) & "', '" & requestArr(i, 2) & "', '" _
            & requestArr(i, 7) & "', '" & peopleInt(3, 2) & "', '" & requestArr(i, 19) & "', '" & requ
estArr(i, 9) & "')"

            peopleInt(2, 2) = AddMinute(peopleInt(2, 2))
            'peopleInt(3, 2) = AddMinute(peopleInt(3, 2)) For BO who was taken out of the approval pro
cess. Keeping in case they bring them back.

            DoCmd.RunSQL strSQL

        End If

        If reqCheck Then

            'tbl_Requirements_and_Program
            strSQL = "INSERT INTO tbl_Requirements_and_Program " _
            & "([Fund Center], Fund, [Funding Source], [Parent Organization], [Extended Organization],
 [Cost Collector], SAG, [Functional Area], AMSCO, MDEP, [Object Class], [Object Class Description]" _
            & ", [Funding Category], Fenced, [Validated Requirement], [Entered By], [Program Update Qu
arter], [Date], Notes, [Mission Criticality])" _
            & " VALUES ('" & requestArr(i, 3) & "', '" & requestArr(i, 6) & "', '" & requestArr(i, 8)
& "', '" & requestArr(i, 4) & "', '" & requestArr(i, 5) & "', '" & requestArr(i, 14) & "', '" _
            & Left(requestArr(i, 15), 3) & "', '" & requestArr(i, 15) & "', '" & Left(requestArr(i, 15
), 6) & "', '" & Right(requestArr(i, 15), 4) & "', '" & requestArr(i, 11) & "', '" _
            & requestArr(i, 12) & "', '" & requestArr(i, 10) & "', '" & requestArr(i, 13) & "', '" & r
equestArr(i, 17) & "', '" & RID & "Tot', '" _
            & requestArr(i, 7) & "', '" & peopleInt(3, 2) & "', '" & requestArr(i, 19) & "', '" & requ
estArr(i, 9) & "')"


            peopleInt(2, 2) = AddMinute(peopleInt(2, 2))
            'peopleInt(3, 2) = AddMinute(peopleInt(3, 2)) For BO who was taken out of the approval pro
cess. Keeping in case they bring them back.

            DoCmd.RunSQL strSQL

        End If

        Dim sPath As String
        Dim dPath As String

        If oExcelWrkBk.Worksheets("Signatures").Cells(16, 2).Value <> "" Then

        'If Worksheets("Signatures").Cells(16, 2).Value <> "" Then
            Set oFS = CreateObject("Scripting.FileSystemObject")
            dPath = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Da
tabase\RequestAttachments\" & RID & typeLabel & " (FY" & FYINFO(Now, "Y") & ")\"
            sPath = oExcelWrkBk.Worksheets("Signatures").Cells(16, 2).Value

            MkDir dPath

            oFS.CopyFile sPath, dPath

        End If

        If oneRecord Then i = i + 1

    Next i

End Sub

Sub excelFormStatus(category As String)

    strDir = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\
1_Data\" & category & " Form"

    Set fso = CreateObject("Scripting.FileSystemObject")

    Set objFiles = fso.GetFolder(strDir).Files

    Count = 0

    For Each objF In objFiles

        If Not objF.Name Like "~$*" Then

            If objF.Name Like "*.xlsm" Then
```

```
                Count = Count + 1

            End If

        End If

    Next objF

    MsgBox "There are " & Count & " " & category & " Forms waiting to be approved or submitted.", , ca
tegory & " Form Status"     'Total number of files

End Sub
```

```
Option Compare Database

Public Function importPgmReq()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databs
e\RTS Database\SOF_Database_Backend\SOF_Database_BE.accdb", False, "g8pae"
'Run Sub procedure.
    'app.Run "updatePgmReq" THIS IS NO LONGER NECESSARY!
Set app = Nothing
End Function
```

```
Option Compare Database

Sub importSpendPlanFile()

    Call updateStatusBar("Importing Spend Plans.....")

    Set fso = CreateObject("Scripting.FileSystemObject")
    sFolder = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database
\1_Data\SpendPlan\Spend Plan Data\"
    fso.CopyFile sFolder & "SpendPlantoImport.xlsm", sFolder & "SpendPlan.xlsm", True

    Call execQuery("000_Delete_SpendPlan")
    Call execQuery("qry_SpendPlan_Append")

    Call clearStatusBar

    MsgBox ("Import Complete")

End Sub

Sub importUnitFundingFile()

    Call updateStatusBar("Importing Unit Funding File.....")

    Set fso = CreateObject("Scripting.FileSystemObject")
    sFolder = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database
\1_Data\Unit Funding\Unit Funding Data\"
    fso.CopyFile sFolder & "UnitFundingtoImport.xlsm", sFolder & "UnitFunding.xlsm", True

    Call execQuery("000_Delete_UnitFunding")
    Call execQuery("qry_UnitFunding_Append")

    Call clearStatusBar

    MsgBox ("Import Complete")

End Sub
```

```
Option Compare Database

Sub importAllSpreadsheets()
SetWarnings = False

    DoCmd.OpenQuery "000_Delete_GFEBS_SOF_VB"
    DoCmd.TransferSpreadsheet acImport, 10, "tbl_GFEBS_SOF_VB", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL
 OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\FY" & FYINFO(Now, "Y") & " GF
EBS SOF - VB.xlsx", True
    DoCmd.OpenQuery "000_Delete_GFEBS_SOF_VV"
    DoCmd.TransferSpreadsheet acImport, 10, "tbl_GFEBS_SOF_VV", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL
 OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\1_Data\Update Reports\FY" & FYINFO(Now, "Y") & " GF
EBS SOF - VV.xlsx", True
    'DoCmd.OpenQuery "000_Delete_ReqsAndPgm"
    'DoCmd.TransferSpreadsheet acImport, 10, "tbl_Requirements_and_Program", "\\RUCKW0U9G67001\drm\PBD
\DATA CALL\PAE\BUDGET_STATUS\FY19 Database Files\FY18 Requirements & Program.xlsx", True, "Source Data
!"
    'DoCmd.OpenQuery "000_Delete_PrgmDetails"
    'DoCmd.TransferSpreadsheet acImport, 10, "tbl_ProgramDetails", "\\RUCKW0U9G67001\drm\PBD\DATA CALL
\PAE\BUDGET_STATUS\FY19 Database Files\FY19 TBG AVNCoE Program.xlsx", True, "Program Details!"

End Sub
```

```
Option Compare Database

Private Const dbLog As Boolean = True 'Set this to False to turn all logging off.

Public Function importVVnVB()
Dim app As Access.Application  'Create instance of Access Application object.
Dim startTime As Date               'To record processing time
Dim endTime As Date                 'To record processing time
Dim execTime As String              'To calculate processing time
Dim dbPath As String                'To capture the current application path
Dim dbName As String                'To capture the current application name
Dim engPath As String               'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()  'record the start time

    Set app = CreateObject("Access.Application")  'Open BE Database in Microsoft Access window.

    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databs
e\RTS Database\2_Engines\SOF_Database_BE\SOF_Database_BE.accdb", False, "g8pae"
'Run Sub procedure.
    app.Run "updateVVnVB"

    endTime = Now()  'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss") 'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "importVVnVB"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

Set app = Nothing
End Function
```

```
Option Compare Database
Option Explicit

'Purpose:       Log for running queries in access engine
'Author:        Brandon Pittmam
'Usage:         Call qLog function when calling queries and fLog in open close events of forms

'Set this to False to turn all logging off.
Private Const dbLog As Boolean = True
'Name of this module (for error logger.)
Private Const conMod = "modLOG"

'API calls to get the Windows user name and computer name
Private Declare PtrSafe Function apiGetUserName Lib "advapi32.dll" _
    Alias "GetUserNameA" (ByVal lpBuffer As String, nSize As Long) As Long

Private Declare PtrSafe Function apiGetComputerName Lib "Kernel32" _
    Alias "GetComputerNameA" (ByVal lpBuffer As String, nSize As Long) As Long

Public Function execQuery(qry As String, Optional enginePath As String, Optional queryParam As String)
 As Long

On Error GoTo Err_Handler
    'Purpose:   Create a log entry for the query being run.
    'Argument:  The query whose execution we are logging.
    'Return:    Primary key value of the log entry. Zero on error.
    'Usage:     For a form, set the On Open property to:    =LogDocOpen([Form])
    '           For a report, set the On Open property to:  =LogDocOpen([Report])
    Dim rs As DAO.Recordset
    Dim lngObjType As Long          'acForm or acReport
    Dim strQry As String            'Name of the query
    Dim startDateTime As Date
    Dim endDateTime As Date
    Dim recordsAffected As Long
    Dim user As String
    Dim compName As String
    Dim dbFullPath As String
    Dim strParam As String
    Dim dbs As DAO.Database
    Dim qdf As DAO.QueryDef
    Dim objAcc As Access.Application
    Dim startTime As Date               'To record processing time
    Dim endTime As Date                 'To record processing time
    Dim execTime As String
    Dim dbPath As String
    Dim dbName As String                'To capture the current application name
    Dim engPath As String               'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName

    If tableExists("TBL_LOG") Then
        Debug.Print "The Table exists"
    Else
        Debug.Print "The table does not exist"
        Debug.Print "Creating Log Table"
        createLogTable
    End If

    strQry = qry
    dbFullPath = enginePath
    strParam = queryParam
    Debug.Print strQry

    If Len(dbFullPath) < 1 Then
        Set dbs = CurrentDb
    Else
        Set objAcc = GetObject(dbFullPath)
        objAcc.Visible = False
        Set dbs = objAcc.CurrentDb

    End If
    Set qdf = dbs.QueryDefs(strQry)
    If Len(strParam) > 0 Then
        Debug.Print strParam
        qdf.Parameters("[" & "Param" & "]") = strParam
        startDateTime = Now()
        startTime = Now()
        qdf.Execute dbFailOnError
        endDateTime = Now()
        endTime = Now()
        recordsAffected = qdf.recordsAffected
    Else
        startDateTime = Now()
        startTime = Now()
        qdf.Execute dbFailOnError
        endDateTime = Now()
        endTime = Now()
        recordsAffected = qdf.recordsAffected
    End If

        user = NetworkUserName()
        'user = Environ("COMPUTERNAME")
        compName = ComputerName()
        'compName = getLogon()
        execTime = Format(endTime - startTime, "hh:mm:ss")
    If dbLog Then

        Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = strQry
            rs!RunDateTime = startDateTime
            rs!EndRunDateTime = endDateTime
            rs!ComputerName = compName
            rs!Username = user
            rs!NumberRecordsAffected = recordsAffected
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
        'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."

        rs.Update
        rs.Bookmark = rs.LastModified
        execQuery = rs!ID
        rs.Close
    End If

Exit_Handler:
    Set rs = Nothing
    Exit Function

Err_Handler:
    Call LogError(Err.Number, Err.Description, conMod & ".execQuery", "Query " & strQry, False)
    Resume Exit_Handler
End Function

Private Function LogError(ByVal lngErrNumber As Long, _
    ByVal strErrDescription As String, _
    strCallingProc As String, _
    Optional vParameters As Variant, _
    Optional bShowUser As Boolean = True) As Boolean
    'Purpose:   Substitute for the real error logging routine at:
    '           http://allenbrowne.com/ser-23a.html

    'If bShowUser Then
        MsgBox "Error " & lngErrNumber & ": " & strErrDescription, vbExclamation, strCallingProc
    'End If
End Function
Private Function ComputerName() As String
On Error GoTo Err_Handler
    'Purpose:   Return the name of this workstation.
    Dim strName As String
    Dim lngLen As Long

    lngLen = 16&
    strName = String$(lngLen, vbNullChar)

    If apiGetComputerName(strName, lngLen) = 0& Then
        ComputerName = "Unknown"
    Else
        ComputerName = Left$(strName, lngLen)
        'Debug.Print ComputerName
    End If

Exit_Handler:
    Exit Function

Err_Handler:
    Call LogError(Err.Number, Err.Description, conMod & ".fOSMachineName")
    Resume Exit_Handler
End Function

Private Function NetworkUserName() As String
On Error GoTo Err_Handler
    'Purpose:   Returns the network login name.
    Dim lngLen As Long          'Length of string.
    Dim strUserName As String
    Const lngcMaxFieldSize As Long = 64& 'Length of field to store this data.

    'Initialize
    strUserName = String$(254, vbNullChar)
    lngLen = 255&

    'API returns a non-zero value if success.
    If apiGetUserName(strUserName, lngLen) <> 0& Then
        lngLen = lngLen - 1&    'Without null termination char.
        If lngLen > lngcMaxFieldSize Then  'Maximum field size
            lngLen = lngcMaxFieldSize
        End If
        NetworkUserName = Left$(strUserName, lngLen)

    End If

Exit_Handler:
    Exit Function

Err_Handler:
    Call LogError(Err.Number, Err.Description, conMod & ".NetworkUserName", , False)
    Resume Exit_Handler
End Function
Function tableExists(tableName As String) As Boolean

Dim strTableNameCheck
On Error GoTo ErrorCode

strTableNameCheck = CurrentDb.TableDefs(tableName)

tableExists = True

ExitCode:
    On Error Resume Next
    Exit Function

ErrorCode:
    Select Case Err.Number
        Case 3265  'Item not found in this collection
            tableExists = False
            Resume ExitCode
        Case Else
            MsgBox "Error " & Err.Number & ": " & Err.Description, vbCritical, "hlfUtils.TableExists"
            'Debug.Print "Error " & Err.number & ": " & Err.Description & "hlfUtils.TableExists"
            Resume ExitCode
    End Select

End Function
Public Function createLogTable()
Dim dbs As DAO.Database
```

```
    Set dbs = CurrentDb
    dbs.Execute "CREATE TABLE TBL_LOG" _
    & "(ID counter, RunDatetime Date, QueryName text,NumberRecordsAffected Int,EndRunDateTime Date, Us
erName Text, ComputerName Text);"
    dbs.Close
End Function
```

```
Option Compare Database


Public Function ListMyFiles(mySourcePath, IncludeSubfolders)

Dim rs As Recordset
Dim file As String
Dim filePath As String
Dim fileName As String
Dim fileModifiedDate As Date
Dim fileCreatedDate As Date
Dim fileSize As Double
Dim myObject As New Scripting.FileSystemObject


    Set mySource = myObject.GetFolder(mySourcePath)
    On Error Resume Next
    For Each myfile In mySource.Files
        filePath = Left(myfile.Path, InStrRev(myfile.Path, "\"))
        Debug.Print Len(myfile.Path)
        Debug.Print InStrRev(myfile.Path, "\")
        fileName = myfile.Name
        fileSize = myfile.Size
        fileModifiedDate = myfile.DateLastModified
        fileCreatedDate = myfile.DateCreated
        Set rs = DBEngine(0)(0).OpenRecordset("TBL_TMP_FILE_INFO", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!FName = fileName
            rs!fPath = filePath
            rs!fCreatedDate = fileCreatedDate
            rs!fModifiedDate = fileModifiedDate
            rs!fSize = fileSize

        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    Next
    If IncludeSubfolders Then
        For Each mySubFolder In mySource.SubFolders
            Call ListMyFiles(mySubFolder.Path, True)
        Next
    End If
End Function
Public Function Copy_Folder(strFrom As String, strTo As String)
'This example copy all files and subfolders from FromPath to ToPath.
'Note: If ToPath already exist it will overwrite existing files in this folder
'if ToPath not exist it will be made for you.
    Dim fso As Object
    Dim FromPath As String
    Dim ToPath As String



    FromPath = strFrom
    ToPath = strTo
    'If you want to create a backup of your folder every time you run this macro
    'you can create a unique folder with a Date/Time stamp.
    'ToPath = "C:\Users\Ron\" & Format(Now, "yyyy-mm-dd h-mm-ss")

    If Right(FromPath, 1) = "\" Then
        FromPath = Left(FromPath, Len(FromPath) - 1)
    End If

    If Right(ToPath, 1) = "\" Then
        ToPath = Left(ToPath, Len(ToPath) - 1)
    End If

    Set fso = CreateObject("scripting.filesystemobject")

    If fso.FolderExists(FromPath) = False Then
        MsgBox FromPath & " doesn't exist"
        Exit Function
    End If
    fso.CopyFolder Source:=FromPath, Destination:=ToPath

    MsgBox "File Transfer Complete"
End Function
Public Function showQueryData()

Dim db As DAO.Database
Dim rs As DAO.Recordset
Dim sqlStr As String

sqlStr = "SELECT * FROM TBL_FILE_INFO as f WHERE f.fFlag=True"

Set db = CurrentDb
Set rs = db.OpenRecordset(sqlStr)

rs.MoveFirst

Do While Not rs.EOF
  Debug.Print (rs!FName)
  rs.MoveNext
Loop

MsgBox ("End of list")

End Function
Public Function AllDescriptions(Optional enginePath As String)
On Error GoTo Err_AllDescriptions

    Dim qdf As QueryDef
    Dim tdf As TableDef
    Dim obj As AccessObject
    Dim varProperty
    Dim strType As String
    Dim dbs As DAO.Database
    Dim dbFullPath As String
    Dim rs As Recordset
    Dim strName As String
    Dim strDesc As String

    dbFullPath = enginePath

    If Len(dbFullPath) < 1 Then
        Set dbs = CurrentDb
    Else
        Set objAcc = GetObject(dbFullPath)
        Set dbs = objAcc.CurrentDb
        Debug.Print objAcc.CurrentProject.FullName
    End If



    ' Queries.
    For Each qdf In dbs.QueryDefs
        If Left(qdf.Name, 1) <> "~" Then
            varProperty = ""
            varProperty = qdf.Properties("Description")
            strName = qdf.Name
            strType = "Query"
            'strDesc = varProperty
            Debug.Print "Query" & "+" & qdf.Name & ":" & varProperty
                        Set rs = DBEngine(0)(0).OpenRecordset("TBL_TMP_DB_OBJECTS", dbOpenDynaset, dbA
ppendOnly)
            rs.AddNew
                rs!ObType = strType
                rs!obName = strName
                'rs!ObDescription = " " & strDesc
                rs!ObDB = objAcc.CurrentProject.FullName
            rs.Update
            rs.Bookmark = rs.LastModified
            rs.Close
        End If
    Next qdf
    ' Tables.
    For Each tdf In CurrentDb.TableDefs
        If Left(tdf.Name, 4) <> "MSys" Then
            varProperty = ""
            varProperty = tdf.Properties("Description")
            ' Set the type based on whether or not the definition
            ' has a connect string, signifying that it is linked.
            If Len(tdf.Connect) > 0 Then
                strType = "Table Link"
            Else
                strType = "Table"
            End If
            Debug.Print strType & "+" & tdf.Name & ":" & varProperty
        End If
    Next tdf

    ' Macros.
    For Each obj In objAcc.CurrentProject.AllMacros
        varProperty = ""
        varProperty = objAcc.CurrentDb.Containers("Scripts").Documents(obj.Name).Properties("Descripti
on")
        Debug.Print "Macro" & "+" & obj.Name & ":" & varProperty
    Next obj


Exit_AllDescriptions:
    On Error Resume Next
    Exit Function

Err_AllDescriptions:
    Select Case Err.Number
    Case 3270
        ' There was no description property for this object.
        Resume Next
    Case Else
        MsgBox Err.Number & " " & Err.Description, vbCritical, "AllDescriptions"
        Resume Exit_AllDescriptions
    End Select

End Function


Public Function RemoteCompactScott2()

Dim SourceFile As String


'SourceFile = "C:\Temp\fmtools\OK_Desktop_Tools\2_Engine\NXG_Engine\NXG_Engine_Active_Year_ROLLOVER.ac
cdb"

'Application.CompactRepair SourceFile, "temp_NXG_Engine_Active_Year_ROLLOVER.accdb"


SourceFile = "C:\Temp\fmtools\OK_Desktop_Tools\2_Engine\FMRP_Engine\FMRP_Engine_Active_Year_ROLLOVER.a
ccdb"

Application.CompactRepair SourceFile, "temp_FMRP_Engine_Active_Year_ROLLOVER.accdb"


End Function

Public Function testLOG()

    Call execQuery("qry_TEST_Log_Make")

End Function
```

```
Option Compare Database

Public Function CompactAndRepairDB()

    Call updateStatusBar("Running Compact and Repair on SOF_Database_BE.....")
    Path = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\2_
Engines\SOF_Database_BE\"
    DBEngine.CompactDatabase Path & "SOF_Database_BE.accdb", Path & "Temp.accdb", , , ";pwd=g8pae"
    Kill Path & "SOF_Database_BE.accdb"
    Name Path & "Temp.accdb" As Path & "SOF_Database_BE.accdb"

    Call updateStatusBar("Running Compact and Repair on PRR_BE.....")
    Path = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Database\2_
Engines\PRR_Engine\PRR_BE\"
    DBEngine.CompactDatabase Path & "PRR_BE.accdb", Path & "Temp.accdb"
    Kill Path & "PRR_BE.accdb"
    Name Path & "Temp.accdb" As Path & "PRR_BE.accdb"

End Function

Public Function RemoteCompact(SourcePath As String, BUPath As String) '
Dim KillFile As String
Dim aFilename As Variant
Dim SourceFile As String
Dim BUFile As String
On Error GoTo RemoteCompact_Err
'These lines assign the variables full path and filenames

SourceFile = SourcePath
BUFile = BUPath
'BUFile = "\\RUCKW0U9G67001\drm\PBD SOF Database\SOF_Database_Backend\BE_Backup\SOF_Database_BE_Backup
.accdb"
'SourceFile = "\\RUCKW0U9G67001\drm\PBD SOF Database\SOF_Database_Backend\SOF_Database_BE.accdb"

'Copies file to backup folder and renames it with the temp_ prefix.
Set aFilename = CreateObject("Scripting.FileSystemObject")
aFilename.CopyFile SourceFile, BUFile, True

'This section deletes the original file if it exists.

'KillFile = SourceFile
'Debug.Print KillFile
Debug.Print BUFile

''Check that file exists
'If Len(Dir$(KillFile)) > 0 Then
    ''First remove readonly attribute, if set
    'SetAttr KillFile, vbNormal
    ''Then delete the file
    'fnWait (5)
    'Kill KillFile
'End If

SetAttr BUFile, vbNormal

'This section copies the temp_ file back to proper location, compacts it, and renames it back to the o
riginal filename.
'DBEngine.CompactDatabase BUFile, SourceFile
FileCopy SourceFile, BUFile

RemoteCompact_Exit:
    Exit Function

RemoteCompact_Err:
    'MsgBox Error$
    Resume RemoteCompact_Exit

End Function
Public Function fnWait(intNrOfSeconds As Integer)
Dim varStart As Variant
  varStart = Timer
  Do While Timer < varStart + intNrOfSeconds
  Loop
```

```
End Function


Public Function RemoteCompactEOY(SourcePath As String, BUPath As String) '
Dim KillFile As String
Dim aFilename As Variant
Dim SourceFile As String
Dim BUFile As String

On Error GoTo RemoteCompactEOY_Err
'These lines assign the variables full path and filenames

SourceFile = SourcePath
BUFile = BUPath

'Copies file to backup folder and renames it with the temp_ prefix.
Set aFilename = CreateObject("Scripting.FileSystemObject")
aFilename.CopyFile SourceFile, BUFile, True

'This section deletes the original file if it exists.

KillFile = SourceFile
Debug.Print KillFile
Debug.Print BUFile

''Check that file exists
If Len(Dir$(KillFile)) > 0 Then
    ''First remove readonly attribute, if set
    SetAttr KillFile, vbNormal
    ''Then delete the file
    fnWait (5)
    Kill KillFile
End If

SetAttr BUFile, vbNormal

'This section copies the temp_ file back to proper location, compacts it, and renames it back to the o
riginal filename.
DBEngine.CompactDatabase BUFile, SourceFile

RemoteCompactEOY_Exit:
    Exit Function

RemoteCompactEOY_Err:
    MsgBox Error$
    Resume RemoteCompactEOY_Exit

End Function
```

```
Option Compare Database

Function addRID(prrType As String) As Integer

    Dim rst As DAO.Recordset
    Dim strSQL As String
    strSQL = "SELECT MAX(INT(MID(" & prrType & ".RID, 1, INSTR(" & prrType & ".RID, ""-"") - 1))) As M
axRID FROM " & prrType & ""
    Set rst = CurrentDb.OpenRecordset(strSQL)

    If Not IsNull(rst!maxRID) Then

        newRID = CInt(rst!maxRID) + 1

    Else

        newRID = 1

    End If

    rst.Close
    Set rst = Nothing

    addRID = newRID

End Function

Function blankRIDQueries()

    Call execQuery("qry_P_BlankRID_Delete")
    Call execQuery("qry_R_BlankRID_Delete")
    Call execQuery("qry_PnR_BlankRID_Delete")

End Function
```

```
Option Compare Database

Sub Auto_Close()
    ThisWorkbook.Saved = True
End Sub
```

```
Option Compare Database

Private Const dbLog As Boolean = True 'Set this to False to turn all logging off.

'------------------------------------------------------------
' SOF Detail export
'
'------------------------------------------------------------
Function exportSOFDetail()
On Error GoTo Macro1_Err
Dim startTime As Date                   'To record processing time
Dim endTime As Date                     'To record processing time
Dim execTime As String                  'To calculate processing time
Dim dbPath As String                    'To capture the current application path
Dim dbName As String                    'To capture the current application name
Dim engPath As String                   'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()  'record the start time

        DoCmd.TransferDatabase acExport, "Microsoft Access", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OP
ERATIONS BRANCH\00 PBD SOF Databse\RTS Database\5_Output\Leadership Database.accdb", acTable, "SOF Det
ail Local", "SOF Detail", False
        DoCmd.TransferDatabase acExport, "Microsoft Access", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OP
ERATIONS BRANCH\00 PBD SOF Databse\RTS Database\5_Output\Leadership Database.accdb", acTable, "qry_CCF
AExceptions", "UnitCCSummary", False
        DoCmd.TransferDatabase acExport, "Microsoft Access", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OP
ERATIONS BRANCH\00 PBD SOF Databse\RTS Database\5_Output\Leadership Database.accdb", acTable, "000 - C
ost Collector Information", "CC_List", False

    endTime = Now()  'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss") 'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "exportSOFDetail"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If

Macro1_Exit:
    Exit Function

Macro1_Err:
    MsgBox Error$
    Resume Macro1_Exit

End Function
```

```
Option Compare Database

Private Const dbLog As Boolean = True 'Set this to False to turn all logging off.

Public Function validateNums()
DoCmd.SetWarnings False
Dim startTime As Date                   'To record processing time
Dim endTime As Date                     'To record processing time
Dim execTime As String                  'To calculate processing time
Dim dbPath As String                    'To capture the current application path
Dim dbName As String                    'To capture the current application name
Dim engPath As String                   'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()  'record the start time

        DoCmd.OpenQuery "qry_SOF_NumValidation_Delete"
        DoCmd.OpenQuery "qry_SOF_NumValidation_VV"
        DoCmd.OpenQuery "qry_SOF_NumValidation_VB"
        'AD no longer used
        'DoCmd.OpenQuery "qry_SOF_NumValidation_AD"

        'Manual button created
        'DoCmd.OpenForm "frm_SOF_NumValidation", acNormal

    endTime = Now()  'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss") 'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "validateNums"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If
    Call clearStatusBar
DoCmd.SetWarnings True
End Function
```

```
Option Compare Database

Private Const dbLog As Boolean = True  'Set this to False to turn all logging off.
'------------------------------------------------------------
'SOF_DB_Update
'
'------------------------------------------------------------
Function SOF_DB_Update()
On Error GoTo SOF_DB_Update_Err

'Dim rs As DAO.Recordset
'Dim currDB As DAO.Database
'Dim rs1 As Recordset
'Set currDB = CurrentDb
'Dim SQL As String

DoCmd.SetWarnings False

    'MsgBox "One moment while the database updates.....", vbOKOnly, "Process Notice"
    Call updateStatusBar("Delete SOF Detail Query.....")
    Call execQuery("000 - Delete SOF Detail Query")

    'Call deleteACCDB

    Call updateStatusBar("Importing GFEBS SOF VV and VB.....")
    importVVnVB  'This function deletes the current GFEBS SOF VV and VB tables from the SOF Backend Da
tabase and imports the (most recent) GFEBS SOF VV and VB into the Backend(BE) SOF Database.  These are
 downloaded from GFEBS during the Morning SOF Update process.

    'Call deleteACCDB
    'MsgBox ("Click ok")

    Call updateStatusBar("FMT - Obligation Log -  Fenced.....")
    Call execQuery("FMT - Obligation Log -  Fenced")

    'Call deleteACCDB

    Call updateStatusBar("FMT - Obligation Log -  Negative Fenced.....")
    Call execQuery("FMT - Obligation Log -  Negative Fenced")

    'Call deleteACCDB

    Call updateStatusBar("VV - SOF Clean Up.....")
    Call execQuery("VV - SOF Clean Up")

    'Call deleteACCDB

    Call updateStatusBar("Requirements & Program w/ Contract Data Query.....")
    Call execQuery("Requirements & Program w/ Contract Data Query")

    'Call deleteACCDB

    Call updateStatusBar("TIGER Program Details Query.....")
    Call execQuery("TIGER Program Details Query")

    'Call deleteACCDB

    Call updateStatusBar("TIGER Phase Obligations Report Query.....")
    Call execQuery("TIGER Obs Report Query")

    'Call deleteACCDB

    Call updateStatusBar("TIGER Phase Commitments Report Query.....")
    Call execQuery("TIGER Comms Report Query")

    Call updateStatusBar("UnitFunding Query.....")
    Call execQuery("UnitFunding Query")

    'Call deleteACCDB

    Call updateStatusBar("VB - SOF Clean Up.....")
    Call execQuery("VB - SOF Clean Up")
    'Call deleteACCDB

    Call updateStatusBar("VV - SOF Clean Up_CivPay.....")
    Call execQuery("VV - SOF Clean Up_CivPay")

    'Call deleteACCDB

    Call updateStatusBar("SOF_Detail_Local_Delete.....")
    Call execQuery("qry_SOF_Detail_Local_Delete")

    'Call deleteACCDB

    Call updateStatusBar("qry_SOF_Detail_Local_Append.....")
    Call execQuery("qry_SOF_Detail_Local_Append")

    'Call deleteACCDB

    'Call updateStatusBar("dlt_tbl_MissingPRRsFromtbl_ReqnPgm....")
    'Call execQuery("dlt_tbl_MissingPRRsFromtbl_ReqnPgm")
    'Call updateStatusBar("qry_MissingPRRsFromtbl_ReqnPgm....")
    'Call execQuery("qry_MissingPRRsFromtbl_ReqnPgm")
    'Call updateStatusBar("UPDATE_MissingfromReqnPgm_PgmNReq_Input....")
    'Call execQuery("UPDATE_MissingfromReqnPgm_PgmNReq_Input")
    'Call updateStatusBar("UPDATE_MissingfromReqnPgm_PgmOnly_Input....")
    'Call execQuery("UPDATE_MissingfromReqnPgm_PgmOnly_Input")
    'Call updateStatusBar("UPDATE_MissingfromReqnPgm_ReqOnly_Input....")
    'Call execQuery("UPDATE_MissingfromReqnPgm_ReqOnly_Input")
    Call updateStatusBar("Updating the Analyst and Leadership databases.....")
    exportSOFDetail ' This function exports the SOF Detail Local table to the Leadership & Analyst dat
abases in the SOF Detail table in both DBs.

    'Call deleteACCDB

    Call updateStatusBar("Validating SOF Detail table record count.....")
   validateSOFDetails

    'Call deleteACCDB

    emailDBUpdate  'This function sends an email to the desiganated recipients (based on their reciept
 status in the "tbl_Emails"), notifying them of an update to the SOF Database.

    'Call deleteACCDB

    'Call updateStatusBar("Generating Morning Numbers validation.....")
    validateNums

    'Call deleteACCDB

    Call clearStatusBar

SOF_DB_Update_Exit:
    Call clearStatusBar
    Exit Function

SOF_DB_Update_Err:
    MsgBox Error$
    Resume SOF_DB_Update_Exit

End Function

Function SOFBackup()

    Call updateStatusBar("Backing up SOF backend database.....")
    backendBackupSOF  'This function creates a backup of the Backend DB and saves it to the designated
 backup folder location on the shared drive.
    Call updateStatusBar("Backing up PRR backend database.....")
    backendPRRBackup  'This function creates a backup of the PRR  Backend DB and saves it to the desig
nated backup folder location on the shared drive.
    Call updateStatusBar("Cleaning up backup folder.....")
    cleanBackups  'After the Backend is backed up on the previous step, this fucntion moves through th
e backup directory and deletes all backups older than 5 days.
    Call clearStatusBar
    Call BackupTime
```

```
End Function

Function deleteACCDB()

    Dim strFileName As String
    Dim strFileExists As String

    'SOF_Database_BE
    strFileName = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\2_Engines\SOF_Database_BE\SOF_Database_BE.laccdb"
    strFileExists = Dir(strFileName)

    If strFileExists <> "" Then
        Kill strFileName
    End If

    'PRR_Engine
    strFileName = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\2_Engines\PRR_Engine\PRR_BE\PRR_BE.laccdb"
    strFileExists = Dir(strFileName)

    If strFileExists <> "" Then
        Kill strFileName
    End If

    'Email_Engine
    strFileName = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\2_Engines\Email_Engine\Email_Engine.laccdb"
    strFileExists = Dir(strFileName)

    If strFileExists <> "" Then
        Kill strFileName
    End If

    'Crosswalk_Engine
    strFileName = "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS Data
base\2_Engines\Crosswalk_Engine\Crosswalk_Engine.laccdb"
    strFileExists = Dir(strFileName)

    If strFileExists <> "" Then
        Kill strFileName
    End If

End Function
```

```
Option Compare Database

Public Function updateStatusBar(Update As String)
Dim RetVal As Variant
RetVal = SysCmd(4, Update)
End Function

Public Function clearStatusBar()
Dim RetVal As Variant
RetVal = SysCmd(5)
End Function
```

```
Option Compare Database

Public Function TDYupdate()
DoCmd.SetWarnings False
    Call updateStatusBar("Updating TDY data....")
    DoCmd.OpenQuery "qry_TDY_Delete"
    DoCmd.OpenQuery "qry_TDY_Append"
    DoCmd.OpenQuery "qry_TDY_Local_Delete"
    DoCmd.OpenQuery "qry_TDY_Local_Append"
    DoCmd.TransferDatabase acExport, "Microsoft Access", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERAT
IONS BRANCH\00 PBD SOF Databse\RTS Database\5_Output\Leadership Database.accdb", acTable, "tbl_TDY_Loc
al", "TDY Analysis", False
    MsgBox "TDY data has been updated.", vbOKOnly, "Update Notice"
    Call clearStatusBar
DoCmd.SetWarnings True
End Function
```

```
Option Compare Database

Public Function UFRdashboard()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD FY " & FYINFO(Now, "Y") & "\1-N report FY" & FYI
NFO(Now, "Y") & "\FY" & FYINFO(Now, "Y") & "_UFR_1_N_BE\FY" & FYINFO(Now, "Y") & "_UFR_1_N_FE_BE.accdb
", False, "USAACE"
'Run Sub procedure.
    app.Run "exportUFR"
Set app = Nothing
End Function
```

ModuleScottTest - 1

```
Option Compare Database

Function fnDmwListAllTables() As String

On Error GoTo errHandler

Dim tbl As AccessObject, db As Object

Dim msg$


Set db = Application.CurrentData

For Each tbl In db.AllTables

Debug.Print tbl.Name

Next tbl

msg$ = "Tables listing complete"


procDone:

fnDmwListAllTables = msg$

Exit Function

errHandler:

msg$ = Err.Number & " " & Err.Description

Resume procDone

End Function
```

```
Option Compare Database

Public Function updateDirDB()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\SOF_Engine\SOF_Engine.acc
db", False
'Run Sub procedure.
    app.Run "exportSOFdata"
Set app = Nothing
End Function

Public Function exportBudgetRpt()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\SOF_Engine\SOF_Engine.acc
db", False
'Run Sub procedure.
    app.Run "copyCharts"
Set app = Nothing
End Function
```

```
Option Compare Database

Public Function updateSOF_Eng()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\SOF_Engine\SOF_Engine.acc
db", False
'Run Sub procedure.
    app.Run "updateSOFCurrent"
Set app = Nothing
End Function

Public Function updateEOM_SOF_Eng()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\SOF_Engine\SOF_Engine.acc
db", False
'Run Sub procedure.
    app.Run "updateEOM"
Set app = Nothing
End Function
Public Function updateEOMrollover()
Dim app As Access.Application
'Create instance of Access Application object.
    Set app = CreateObject("Access.Application")
'Open BE Database in Microsoft Access window.
    app.OpenCurrentDatabase "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\SOF_Engine\SOF_Engine.acc
db", False
'Run Sub procedure.
    app.Run "rolloverEOM"
Set app = Nothing
End Function
```

```
Option Compare Database
Private Const dbLog As Boolean = True 'Set this to False to turn all logging off.

Public Function validateSOFDetails()
Dim strSQL As String
Dim strSQL1 As String
Dim db As Database
Dim rs As Recordset
Dim rs1 As Recordset
Dim rCount As Integer
Dim startTime As Date                'To record processing time
Dim endTime As Date                  'To record processing time
Dim execTime As String               'To calculate processing time
Dim dbPath As String                 'To capture the current application path
Dim dbName As String                 'To capture the current application name
Dim engPath As String                'To combine the current application path and name

    dbName = Application.CurrentProject.Name
    dbPath = Application.CurrentProject.Path
    engPath = dbPath & "\" & dbName
    startTime = Now()  'record the start time

    Set db = CurrentDb
    strSQL = "SELECT Count([SOF Detail Local].[Transaction Source]) AS LRecCount FROM [SOF Detail Loca
l];"
    strSQL1 = "SELECT Count([SOF Detail].[Transaction Source]) AS RecCount FROM [SOF Detail];"
    Set rs = CurrentDb.OpenRecordset(strSQL)
    Set rs1 = CurrentDb.OpenRecordset(strSQL1)

        If rs1.Fields(0).Value - rs.Fields(0).Value <> 0 Then
            MsgBox "SOF Detail table variation is: " & rs1.Fields(0).Value - rs.Fields(0).Value, vbOKO
nly
        Else
            MsgBox "The SOF Detail connected and local table's record counts are equal.", vbOKOnly
        End If

    endTime = Now()  'record the end time
    execTime = Format(endTime - startTime, "hh:mm:ss") 'calculate the processing time
    'Log the processing time and DB information
    If dbLog Then
    Set rs = DBEngine(0)(0).OpenRecordset("TBL_LOG", dbOpenDynaset, dbAppendOnly)
        rs.AddNew
            rs!QueryName = "validateSOFDetails"
            rs!RunDateTime = startTime
            rs!EndRunDateTime = endTime
            rs!ExecutionTime = execTime
            rs!DatabasePath = engPath
            'Debug.Print "Processing time was: " & Format(endTime - startTime, "hh:mm:ss") & "."
        rs.Update
        rs.Bookmark = rs.LastModified
        rs.Close
    End If
End Function
```

```
Option Compare Database

Function closeoutEOY()
DoCmd.SetWarnings False
    yearAdd = IIf(Month(Now) < 10, 1, 0)
    Call updateStatusBar("Making copies of files in \\RUCKW0U9G67001\drm\PBD\DATA CALL\PAE\BUDGET_STAT
US\FY## Database Files.....")
    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\2_Engines\SOF_Database_BE\SOF_Database_BE.accdb", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPE
RATIONS BRANCH\00 PBD SOF Databse\RTS Database\Archive\SOF_Database_BE_EOY" & PriorFY(DateAdd("yyyy",
yearAdd, Now), "Y") & ".accdb")
    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\2_Engines\PRR_Engine\PRR_BE\PRR_BE.accdb", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS
 BRANCH\00 PBD SOF Databse\RTS Database\" & "PRR_BE_EOY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y")
 & ".accdb")
    Call RemoteCompact("\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANCH\00 PBD SOF Databse\RTS
 Database\5_Output\Leadership Database.accdb", "\\RUCKW0U9G67001\drm\PBD\00 FINANCIAL OPERATIONS BRANC
H\00 PBD SOF Databse\RTS Database\" & "Leadership Database_EOY" & PriorFY(DateAdd("yyyy", yearAdd, Now
), "Y") & ".accdb")
    DoCmd.OutputTo acOutputTable, "FMT - Obligation Log - Fenced", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U
9G67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") & " Data
base Files\FMT - Obligation Log - Fenced.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "FMT - Obligation Log - Negative Fenced", "ExcelWorkbook(*.xlsx)", "
\\RUCKW0U9G67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y")
 & " Database Files\FMT - Obligation Log - Negative Fenced.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "Functional Area Information", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G
67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") & " Databa
se Files\Functional Area Information.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "Parent Extended and Branch Info", "ExcelWorkbook(*.xlsx)", "\\RUCKW
0U9G67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") & " Da
tabase Files\Parent Extended and Branch Info.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "TIGER Program Details", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G67001\
drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") & " Database Fil
es\TIGER Program Details.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "tbl_Requirements_and_Program", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9
G67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") & " Datab
ase Files\Requirements & Program.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "Published_Budget_Tiger", "ExcelWorkbook(*.xlsx)", "\\RUCKW0U9G67001
\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") & " Database Fi
les\Published_Budget_Tiger.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "06_tbl_ParentExtended_and_BranchInfo", "ExcelWorkbook(*.xlsx)", "\\
RUCKW0U9G67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") &
 " Database Files\OrgChart.xlsx", False, "", , acExportQualityPrint
    DoCmd.OutputTo acOutputTable, "02_tbl_CostCollectorInformation", "ExcelWorkbook(*.xlsx)", "\\RUCKW
0U9G67001\drm\PBD\DATA CALL\PAE\BUDGET_STATUS\FY" & PriorFY(DateAdd("yyyy", yearAdd, Now), "Y") & " Da
tabase Files\CostCollector.xlsx", False, "", , acExportQualityPrint

    Call updateStatusBar("Deleting old data from SOF tables.....")
    DoCmd.OpenQuery "000_Delete_TBL_LOG"
    Call execQuery("000 - Delete SOF Detail Query")
    Call execQuery("qry_SOF_Detail_Local_Delete")
    Call execQuery("000_Delete_GFEBS_SOF_VV")
    Call execQuery("000_Delete_GFEBS_SOF_VB")
    Call execQuery("000_Delete_ReqsAndPgm")
    Call execQuery("000_Delete_FMT_Fenced")
    Call execQuery("000_Delete_FMT_Neg_Fenced")
    Call execQuery("000_Delete_PrgmDetails")
    Call updateStatusBar("Deleting old data from Program and Requirements tables.....")
    'Call execQuery("000_qry_All_Requests_Delete")
    'Call execQuery("000_qry_ArchivedRequests_Delete")
    Call execQuery("qry_PgmMonthly_Delete")
    Call execQuery("qry_PnR_Delete")
    Call execQuery("qry_ReqOnly_Delete")

    'Call CompactAndRepairDB

    'Call updateStatusBar("Running Compact and Repair on SOF_Database_BE.....")
    'Path = "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\SOF_Database_BE\"
    'DBEngine.CompactDatabase Path & "SOF_Database_BE.accdb", Path & "Temp.accdb", , , ";pwd=g8pae"
    'Kill Path & "SOF_Database_BE.accdb"
    'Name Path & "Temp.accdb" As Path & "SOF_Database_BE.accdb"

    'Call updateStatusBar("Running Compact and Repair on PRR_BE.....")
    'Path = "\\RUCKW0U9G67001\drm\PBD SOF Database\2_Engines\PRR_Engine\PRR_BE\"
    'DBEngine.CompactDatabase Path & "PRR_BE.accdb", Path & "Temp.accdb"
    'Kill Path & "PRR_BE.accdb"
    'Name Path & "Temp.accdb" As Path & "PRR_BE.accdb"

    Call clearStatusBar
    DoCmd.Close acForm, "frm_Closeout"
    MsgBox "EOY closeout is complete.", vbOKOnly, "Update Notice"
    DoCmd.OpenForm "frm_MainControl", acNormal
DoCmd.SetWarnings True
End Function

Sub dateTest()

yearAdd = IIf(Month(Now) < 8, 1, 0)

dateVar = PriorFY(DateAdd("yyyy", yearAdd, Now), "Y")

End Sub
```

```
Option Compare Database

Public Function GetSQLFromQuery(strQueryName As String) As String

    'Runs by opening zqry_GetAllSQL
    'Creates a list of all the queries and the SQL (good for searching queries for certain things)

    Dim db As DAO.Database

    Dim qd As DAO.QueryDef

    Set db = CurrentDb

    Set qd = db.QueryDefs(strQueryName)

    GetSQLFromQuery = qd.SQL

    Set qd = Nothing

    Set db = Nothing

End Function
```