

```
Module mdlMain
```

```
Public strCompany As String = "Tricera-Shop"  
Public blnUseCurrMonth As Boolean = True
```

```
'Application Paths, Etc  
Public strAppPath = System.AppDomain.CurrentDomain.BaseDirectory()  
Public dbsource = "Data Source =" & System.AppDomain.CurrentDomain.BaseDirectory() &  
"/dbSales.mdb"  
Public dbSpeechSource = "Data Source =" &  
System.AppDomain.CurrentDomain.BaseDirectory() & "/Resources/dbSpeech.mdb"  
Public dbAccountsSource = "Data Source =" &  
System.AppDomain.CurrentDomain.BaseDirectory() & "/dbAccounts.mdb"
```

```
Function GetUserName() As String  
If TypeOf My.User.CurrentPrincipal Is _  
Security.Principal.WindowsPrincipal Then  
    ' The application is using Windows authentication.  
    ' The name format is DOMAIN\USERNAME.  
    Dim parts() As String = Split(My.User.Name, "\")  
    Dim username As String = parts(1)  
    Return username  
Else  
    ' The application is using custom authentication.  
    Return My.User.Name  
End If  
End Function
```

```
Public Function WithCurrencySign(strValue As String, blnShow As Boolean) As String  
  
    'Check If Value Already Has A Currency Sign  
    If InStr(strValue, "£") > 0 Then  
        If blnShow = True Then  
            'Leave £ Sign In  
            WithCurrencySign = strValue  
        Else  
            'Remove £ Sign  
            Dim intLen As Integer  
            intLen = Len(strValue)  
            WithCurrencySign = Mid(strValue, 2, intLen - 1)  
        End If  
    Else  
        If blnShow = True Then  
            'Add £ Sign  
            WithCurrencySign = "£" & strValue  
        Else  
            'Leave £ Sign Out  
            WithCurrencySign = strValue  
        End If  
    End If  
End Function
```

End Function

```
Public Function GetSupplierName(strSupplierCode As String) As String
```

```
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select * from tblSuppliers Where SupplierCode = '" & strSupplierCode & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblStock")

    'Check For Null Return

    GetSupplierName = ds.Tables("tblStock").Rows(0).Item("Name")

    con.Close()
```

End Function

```
Public Function GetSupplierCode(strSupplierName As String) As String
```

```
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select * from tblSuppliers Where Name = '" & strSupplierName & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblSuppliers")
    GetSupplierCode = ""
    'Check For Null Return
    If ds.Tables("tblSuppliers").Rows.Count > 0 Then
        GetSupplierCode = ds.Tables("tblSuppliers").Rows(0).Item("SupplierCode")
    End If

    con.Close()
```

End Function

```
Public Sub FillDataList(ByRef objList As Object, ByVal strSQL As String)
```

```
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = strSQL
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "MyData")
    Dim MaxRecCount As Integer
    Dim inc As Integer
    inc = -1
    MaxRecCount = ds.Tables("MyData").Rows.Count
    For inc = 0 To MaxRecCount - 1
        'Dont Add Empty Options !!!
        If (ds.Tables("MyData").Rows(inc).Item(0)) IsNot DBNull.Value Then
            objList.Items.Add(ds.Tables("MyData").Rows(inc).Item(0))
        End If
    Next inc
    con.Close()
```

```
End Sub
```

```
Public Function CreateNewStockNumber() As String
```

```
    On Error GoTo Err_Fail
    'Open Database
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim intNewStockNum As Integer

    'Select Highest Current Session ID
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "SELECT top 1 right(StockCode,8) as StockID from tblStock where
left(StockCode,1)='S' order by right(StockCode,8) desc"

    da = New OleDb.OleDbDataAdapter(sql, con)
```

```

' Check For No Records
da.Fill(ds, "StockID")
If ds.Tables("StockID").Rows.Count = 0 Then
    CreateNewStockNumber = 1
Else
    'Increment highest ID And Return It
    intNewStockNum = ds.Tables("StockID").Rows(0).Item("StockID") + 1
End If
con.Close()

'Format New Stock Code Number (S00000500)
CreateNewStockNumber = intNewStockNum.ToString()
CreateNewStockNumber = "S" & CreateNewStockNumber.PadLeft(8, "0")

Exit Function
Err_Fail:
    MsgBox("Error Creating New Session ID. (This Is A Critical Error)",
MsgBoxStyle.Critical)
    CreateNewStockNumber = ""
End Function

Public Function IsAlreadyPacked(strOrderNum As String) As Boolean

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select * from tblDeliveryAdds Where OrderNum = '" & strOrderNum & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "myData")

    If ds.Tables("myData").Rows(0).Item("AlreadyPacked") = True Then
        IsAlreadyPacked = True
    Else
        IsAlreadyPacked = False
    End If

    con.Close()

End Function

Public Function IsCancelled(strOrderNum As String) As Boolean

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"

```

```

'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()

Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
'''
sql = "select Cancelled from tblDeliveryAdds Where OrderNum = '" & strOrderNum &

da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "myData")

If ds.Tables("myData").Rows(0).Item("Cancelled") = True Then
    IsCancelled = True
Else
    IsCancelled = False
End If

con.Close()

```

End Function

Public Function GetStockCodeFromDesc(strShortDesc As String) As String

```

Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()

Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "select * from tblStock Where Shortdesc = '" & strShortDesc & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "tblStock")

'Check For Null Return

GetStockCodeFromDesc = ds.Tables("tblStock").Rows(0).Item("StockCode")

con.Close()

```

End Function

Public Function GetStockItemDescription(strStockCode As String) As String

```

Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"

```

```

'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()

Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "select Description from tblStock Where StockCode = '" & strStockCode & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "tblStock")

'Check For Null Return
If ds.Tables("tblStock").Rows.Count > 0 Then
    GetStockItemDescription = ds.Tables("tblStock").Rows(0).Item("Description")
Else
    GetStockItemDescription = ""

End If

con.Close()

```

End Function

Public Function GetSHORTStockItemDescription(strStockCode As String) As String

```

Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()

Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "select ShortDesc from tblStock Where StockCode = '" & strStockCode & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "tblStock")

'Check For Null Return
If ds.Tables("tblStock").Rows.Count > 0 Then
    GetSHORTStockItemDescription =
ds.Tables("tblStock").Rows(0).Item("ShortDesc")
Else
    GetSHORTStockItemDescription = ""

End If

con.Close()

```

End Function

```

Public Function CreateNewORDERNumber() As String
    On Error GoTo Err_Fail
    'Open Database
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim intNewOrderNum As Integer
    'Select Highest Current Order Number
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "SELECT top 1 right(OrderNum,8) as OrderID from tblOrders order by
right(OrderNum,8) desc"
    da = New OleDb.OleDbDataAdapter(sql, con)
    ' Check For No Records
    da.Fill(ds, "OrderNum")
    If ds.Tables("OrderNum").Rows.Count = 0 Then
        CreateNewORDERNumber = 1
    Else
        'Increment highest ID And Return It
        intNewOrderNum = ds.Tables("OrderNum").Rows(0).Item("OrderID") + 1
    End If
    con.Close()
    'Format New Order Number (ORD00000500)
    CreateNewORDERNumber = intNewOrderNum.ToString()
    CreateNewORDERNumber = "ORD" & CreateNewORDERNumber.PadLeft(8, "0")
    Exit Function
Err_Fail:
    MsgBox("Error Creating New Order ID. (This Is A Critical Error)",
MsgBoxStyle.Critical)
    CreateNewORDERNumber = ""
End Function

```

```

Public Function GetPostageCharge(strStockCode As String, blnUK As Boolean) As Double

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    'Decide If Postage C0st Is UK Or International
    If blnUK = True Then
        sql = "select Standard_Post_Charge_UK as CHARGE from tblStock Where StockCode
= '' & strStockCode & ''"
    Else

```

```

        sql = "select Standard_Post_Charge_AIR as CHARGE from tblStock Where
StockCode = '" & strStockCode & "'"
    End If

    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblStock")

    'Check For Null Return
    If ds.Tables("tblStock").Rows.Count > 0 Then
        GetPostageCharge = ds.Tables("tblStock").Rows(0).Item("CHARGE")
    Else
        GetPostageCharge = 0.0
    End If

    con.Close()

```

End Function

```

Public Function GetGridTotal(objGrid As Object, intcol As Integer) As Double
    On Error GoTo Err_Fail
    Dim X As Integer
    GetGridTotal = 0
    Dim currValue As Double
    'Loop Through Datagrid
    For X = 0 To objGrid.RowCount - 1
        'Protect From NULL Values In Grid
        If (objGrid.Rows(X).Cells(intcol).value) IsNot DBNull.Value Then
            currValue =
Val(WithCurrencySign(Trim(objGrid.Rows(X).Cells(intcol).Value), False))
            If currValue <> 0 Then
                GetGridTotal = GetGridTotal + currValue
            End If
        End If
    Next X
    Exit Function
Err_Fail:
    GetGridTotal = 0
    MsgBox(ErrorToString)
End Function

```

Public Function GetCustomerCodeFromOrder(strOrderCode As String) As String

```

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String

```



```

... sql = "select CustomerCode from tblOrders Where Ordernum = '" & strOrderCode &
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "tblOrder")
'Check For Null Return

GetCustomerCodeFromOrder = ds.Tables("tblOrder").Rows(0).Item("CustomerCode")
con.Close()

```

End Function

Public Function GetCustomerNameFromCustCode(strCustCode As String) As String

```

Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "select Name from tblCustomers Where CustomerCode = '" & strCustCode & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "tblCusts")
'Check For Null Return

GetCustomerNameFromCustCode = ds.Tables("tblCusts").Rows(0).Item("Name")
con.Close()

```

End Function

Public Function IsOrderComplete(strOrderNum As String) As Boolean

```

Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "select Paid from tblOrders Where OrderNum = '" & strOrderNum & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "myData")

'Check If 'Paid' Date Has Been Entered
If IsDate(ds.Tables("myData").Rows(0).Item("Paid")) = True Then
    IsOrderComplete = True
Else
    IsOrderComplete = False
End If
con.Close()

```

End Function

```
Public Function GetOriginalOrderDate(strOrderCode As String) As String
```

```
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select OriginalDate from tblOrders Where Ordernum = '" & strOrderCode &
    """"

    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblOrder")
    'Check For Null Return

    GetOriginalOrderDate = ds.Tables("tblOrder").Rows(0).Item("OriginalDate")
    con.Close()
```

```
End Function
```

```
Public Function GetPostDate(ByVal strOrderCode As String) As String
```

```
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select DispatchDate from tblDeliveryAdds Where OrderNum = '" &
    strOrderCode & """"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblDelivery")
    'Check For Null Return

    GetPostDate = ds.Tables("tblDelivery").Rows(0).Item("DispatchDate")
    con.Close()
```

```
End Function
```

```
Public Function GetOrderCompany(strOrderCode As String) As String
```

```
    On Error GoTo Err_Fail
```

```
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
```

```

con.ConnectionString = dbprovider & dbsource
con.Open()
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "select Company from tblOrders Where Ordernum = '" & strOrderCode & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "tblOrder")
'Check For Null Return

If ds.Tables("tblOrder").Rows.Count > 0 Then
    GetOrderCompany = ds.Tables("tblOrder").Rows(0).Item("Company")
Else
    GetOrderCompany = "..."
End If

con.Close()
Exit Function
Err_Fail:
    GetOrderCompany = "..."
con.Close()
End Function

Public Function NextWorkingDay() As String
    On Error GoTo Err_Fail
    '#####
    'Include Check For Bank Holidays...
    NextWorkingDay = ""
    '#####
    Dim X As Integer
    For X = 1 To 7
        If IsWorkingDay(Format(DateAdd(DateInterval.Day, X, Date.Today), "ddd")) =
True Then
            'Check This Date For A Bank Holiday Date ??? (If Bank Holiday Move To
Next Day, etc)
            NextWorkingDay = DateAdd(DateInterval.Day, X, Date.Today)
            Exit For
        End If
    Next
    Exit Function
Err_Fail:
    NextWorkingDay = ""
    MsgBox(ErrorToString)
End Function

Public Function ReturnRecordCount(strSQL As String) As Integer
    On Error GoTo Err_Fail

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet

```

```

Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = strSQL
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "MyData")
Dim MaxRecCount As Integer
Dim inc As Integer
inc = -1
ReturnRecordCount = ds.Tables("MyData").Rows.Count
con.Close()
Exit Function
Err_Fail:
ReturnRecordCount = 0
End Function

Public Function ReturnRecordValueSUM(strSQL As String, strField As String) As Long
On Error GoTo Err_Fail

Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = strSQL
da = New OleDb.OleDbDataAdapter(sql, con)

da.Fill(ds, "MyData")

Dim MaxRecCount As Integer
Dim inc As Integer
inc = -1
MaxRecCount = ds.Tables("MyData").Rows.Count
Dim currTotal As Double = 0

For inc = 0 To MaxRecCount - 1

'Dont Add Empty Options !!!
If (ds.Tables("MyData").Rows(inc).Item(0)) IsNot DBNull.Value Then

currTotal = currTotal + ds.Tables("MyData").Rows(inc).Item(strField)

End If
Next inc

ReturnRecordValueSUM = currTotal
con.Close()

Exit Function
Err_Fail:
ReturnRecordValueSUM = 1
End Function

```

```

Public Function IsPostCode(strPCode As String) As Boolean
    On Error GoTo Err_Fail
    'Protect Against Empty String
    If Len(strPCode) < 5 Then
        IsPostCode = False
        Exit Function
    End If

    IsPostCode = False
    'Remove Spaces
    strPCode = strPCode.Replace(" ", "")
    ' Check For Formats > 5,6 or 7
    If Len(strPCode) > 4 Or Len(strPCode) < 8 Then
        'First And Last Chars Should Be Alpha
        If IsNumeric(Mid(strPCode, 1, 1)) = False And IsNumeric(Mid(strPCode,
Len(strPCode), 1)) = False Then
            'Third Digit Should Be Numeric (With One Exception)
            If IsNumeric(Mid(strPCode, 3, 1)) = True Then
                IsPostCode = True
            Else
                'Test London Based Postcode Option
                If IsNumeric(Mid(strPCode, 4, 1)) = True Then IsPostCode = True
            End If
        End If
    End If
    Exit Function
Err_Fail:
    MsgBox(ErrorToString)
    IsPostCode = False
End Function

```

```

Public Function FormatPostCode(strPCode As String) As String
    On Error GoTo Err_Fail
    FormatPostCode = strPCode
    'Remove Spaces
    strPCode = strPCode.Replace(" ", "")

    'Convert To UpperCase
    strPCode = StrConv(strPCode, VbStrConv.Uppercase)

    Select Case Len(strPCode)
        Case 5
            FormatPostCode = Mid(strPCode, 1, 2) & " " & Mid(strPCode, 3, 3)
        Case 6
            FormatPostCode = Mid(strPCode, 1, 3) & " " & Mid(strPCode, 4, 3)
        Case 7
            FormatPostCode = Mid(strPCode, 1, 4) & " " & Mid(strPCode, 5, 3)
        Case Else
            FormatPostCode = strPCode
    End Select

    Exit Function
Err_Fail:
    MsgBox(ErrorToString)
    FormatPostCode = strPCode
End Function

```

```

Public Function GetSuggestedTitle(strName As String) As String
    On Error GoTo Err_Fail
    GetSuggestedTitle = ""
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "SELECT top 5 tblDeliveryadds.title, tblDeliveryadds.FirstName,
Count(tblDeliveryadds.FirstName) as MYCOUNT From tblDeliveryadds WHERE
tblDeliveryadds.FirstName = '" & strName & "' GROUP BY tblDeliveryadds.FirstName,
tblDeliveryadds.Title ORDER BY Count(tblDeliveryadds.FirstName) DESC"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "MyData")
    'Check For Null Return
    If (ds.Tables("MyData").Rows(0).Item("title")) IsNot DBNull.Value Then
        GetSuggestedTitle = ds.Tables("MyData").Rows(0).Item("title")
    Else
        GetSuggestedTitle = ""
    End If
    con.Close()

    Exit Function
Err_Fail:
    con.Close()

End Function

```

```

Public Function CreateNewTransactionNumber() As Long
    On Error GoTo Err_Fail

    'Open Database
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\Accounts.mdb"
    con.ConnectionString = dbprovider & dbAccountsSource
    con.Open()

    'Select Highest Current Session ID
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "SELECT top 1 ID from tbl_Transactions order by ID desc"
    da = New OleDb.OleDbDataAdapter(sql, con)

    ' Check For No Records
    da.Fill(ds, "Transactions")
    If ds.Tables("Transactions").Rows.Count = 0 Then

```

```

        CreateNewTransactionNumber = 1
    Else
        'Increment Highest ID And Return It
        CreateNewTransactionNumber = ds.Tables("Transactions").Rows(0).Item("ID") + 1
    End If
    con.Close()

    Exit Function
Err_Fail:
    MsgBox("Error Creating New Transaction ID. (This Is A Critical Error)",
MsgBoxStyle.Critical)
    CreateNewTransactionNumber = 0
    con.Close()
End Function

Public Function IsByCourier(strOrderNum As String) As Boolean

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select * from tblDeliveryAdds Where OrderNum = '" & strOrderNum & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "myData")
    'Post_Method

    If ds.Tables("myData").Rows(0).Item("Post_Method") = "Courier" Then
        IsByCourier = True
    Else
        IsByCourier = False
    End If

    con.Close()

End Function

Public Function IsValidOrderNumber(strOrderNum As String) As Boolean
    On Error GoTo Err_Fail
    IsValidOrderNumber = False

    If InStr(strOrderNum, "ORD") = 1 Then

        If Len(strOrderNum) = 11 Then

            IsValidOrderNumber = True

```

```

        End If

    End If

    Exit Function
Err_Fail:
    MsgBox(ErrorToString.ToString)
End Function

Public Function IsWorkingDay(strDayName As String) As Boolean
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select * from tblSettings Where DayName = '" & strDayName & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "myData")
    IsWorkingDay = ds.Tables("myData").Rows(0).Item("Working") = True
    con.Close()
End Function

Public Function MarkAsAlreadyPacked(ByVal strOrderNum As String) As Boolean

    'Trim tracking Code If There Is One
    If Len(strOrderNum) <> 11 Then
        MarkAsAlreadyPacked = False
        Exit Function
    End If

    'Make Required Declarations...
    Dim sql As String
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    Dim upd As New OleDb.OleDbCommand
    con.Open()
    sql = "UPDATE tblDeliveryAdds SET [AlreadyPacked]=TRUE WHERE OrderNum = '" &
strOrderNum & "'"
    Try
        upd = New OleDb.OleDbCommand(sql, con)
        upd.ExecuteNonQuery()
        con.Close()
        upd.Dispose()
    End Try

```



```

        MarkAsAlreadyPacked = True
    Catch ex As Exception
        MarkAsAlreadyPacked = False
        MsgBox.Show("Couldnt Update Order Packed Status..." & Err.Description,
"Caution")
        con.Close()
        upd.Dispose()
    End Try
End Function

```

```

Public Function GetTrackingNumber(ByVal strOrderCode As String) As String

```

```

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "select TrackingRef from tblDeliveryAdds Where OrderNum = '" & strOrderCode
& "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblTracking")
    'Check For Null Return

    GetTrackingNumber = ds.Tables("tblTracking").Rows(0).Item("TrackingRef")
    con.Close()

End Function

```

```

Public Function GetOrderEmailGreeting(ByVal strOrderCode As String) As String

```

```

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String

    sql = "select Title, FirstName, Surname, eBayName from tblDeliveryAdds Where
OrderNum = '" & strOrderCode & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblGreeting")

    'Check For Null Return

    If Len(ds.Tables("tblGreeting").Rows(0).Item("eBayName")) > 0 Then
        'Use Ebay Name If There Is One...
    End If

```

```

        GetOrderEmailGreeting = "Dear " &
ds.Tables("tblGreeting").Rows(0).Item("eBayName")
    Else
        'Otherwise Use Full Name (With Title If There Is One...)
        If Len(ds.Tables("tblGreeting").Rows(0).Item("Title")) > 0 Then
            GetOrderEmailGreeting = "Dear " &
ds.Tables("tblGreeting").Rows(0).Item("Title") & " " &
ds.Tables("tblGreeting").Rows(0).Item("FirstName") & " " &
ds.Tables("tblGreeting").Rows(0).Item("Surname")
        Else
            GetOrderEmailGreeting = "Dear " &
ds.Tables("tblGreeting").Rows(0).Item("FirstName") & " " &
ds.Tables("tblGreeting").Rows(0).Item("Surname")
        End If
    End If

    GetOrderEmailGreeting = GetOrderEmailGreeting & Chr(13) & Chr(13)

con.Close()

```

End Function

```

Public Function IsEbayAddressCode(ByVal streBayCode As String) As Boolean
    On Error GoTo Err_Fail

    IsEbayAddressCode = True

    'Trim Spaces
    streBayCode = streBayCode.Trim()

    'Check Length & Protect Against Empty String
    If Len(streBayCode) <> 11 Then
        IsEbayAddressCode = False
        Exit Function
    End If

    'Check For 'eBay' String
    If InStr(streBayCode, "ebay", CompareMethod.Text) <> 1 Then
        IsEbayAddressCode = False
        Exit Function
    End If

    'Check For Spaces
    If InStr(streBayCode, " ", CompareMethod.Text) <> 0 Then
        IsEbayAddressCode = False
        Exit Function
    End If

    Exit Function

```

Err\_Fail:

```
        MsgBox(ErrorToString)
        IsEbayAddressCode = False
    End Function
```

```
End Module
```

```
Imports System.Data.OleDb
```

```
Partial Public Class mdiShell
```

```
    Dim provider As String
    Dim dataFile As String
    Dim connString As String
    Public myConnection As OleDbConnection = New OleDbConnection
    Public dr As OleDbDataReader
    Dim DBNull As Boolean
```

```
    Private Sub ViewStockToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles ViewStockToolStripMenuItem.Click
        frmStock.MdiParent = Me
        frmStock.Show()
    End Sub
```

```
    Private Sub OutstandingToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles OutstandingToolStripMenuItem.Click
        frmOutstanding.MdiParent = Me
        frmOutstanding.Show()
    End Sub
```

```
    Private Sub QUITToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles QUITToolStripMenuItem.Click
```

```
        If bInSPEECH = True Then
```

```
            strMTSpeechText = GetSPEECH("Goodbye", "Anytime", "n")
```

```
            Dim thread As System.Threading.Thread
            thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
            thread.Start()
```

```
        End If
```

```
        Application.Exit()
    End Sub
```

```

    Private Sub CreateNewToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles CreateNewToolStripMenuItem.Click
        frmOrders.MdiParent = Me
        frmOrders.Show()
    End Sub

    Private Sub OutstandingToolStripMenuItem1_Click(sender As System.Object, e As
System.EventArgs) Handles OutstandingToolStripMenuItem1.Click
        frmOutstanding.MdiParent = Me
        frmOutstanding.Show()
    End Sub

    Private Sub OrdersToolStripMenuItem1_Click(sender As System.Object, e As
System.EventArgs) Handles OrdersToolStripMenuItem1.Click
        frmOrders.MdiParent = Me
        frmOrders.Show()
    End Sub

    Private Sub StockToolStripMenuItem1_Click(sender As System.Object, e As
System.EventArgs) Handles StockToolStripMenuItem1.Click
        frmStock.MdiParent = Me
        frmStock.Show()
    End Sub

    Private Sub QUITToolStripMenuItem3_Click(sender As System.Object, e As
System.EventArgs) Handles QUITToolStripMenuItem3.Click

        If blnSPEECH = True Then
            'Dim SAPI
            'SAPI = CreateObject("SAPI.spvoice")
            'SAPI.Speak(GetSPEECH("Goodbye", "Anytime", "n"))

            strMTSpeechText = GetSPEECH("Goodbye", "Anytime", "n")

            Dim thread As System.Threading.Thread
            thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
            thread.Start()

        End If

        Application.Exit()
    End Sub
    Private Sub mdiShell_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load

        blnUseCurrMonth = False

        'Open Orders Outstanding Form By Default
        frmOutstanding.MdiParent = Me
        frmOutstanding.Show()

        'Build Menu list Of Linked Documents
        BuildLinkedDocsMenuList()

        'Make Sure Form Loads Before Speech Interrupts
        Application.DoEvents()

```

```

#####
'Check If Speech Is Enabled...
blnSPEECH = IsSpeechOn()

'May Not Be Needed..?
'Me.CheckForIllegalCrossThreadCalls = False

If blnSPEECH = True Then
    'SPEECH SYTHESIS

    ' Get Hour (24Hour)
    Dim intHour As Integer = DateTime.Now.ToString("HH")
    Dim strGreeting As String
    'Decide What Greeting Is Required
    Select Case intHour
        Case 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
            strGreeting = GetSPEECH("Greeting", "Morning", "n")
        Case 12, 13, 14, 15, 16, 17
            strGreeting = GetSPEECH("Greeting", "Afternoon", "n")
        Case 18, 19, 20, 21
            strGreeting = GetSPEECH("Greeting", "Evening", "n")
        Case 22, 23
            strGreeting = GetSPEECH("Greeting", "Night", "n")
        Case Else
            strGreeting = "Hi There"
    End Select

    'Initialise Speech Sythesis
    'Dim SAPI
    'SAPI = CreateObject("SAPI.spvoice")
    'SAPI.Speak(strGreeting)

    strMTSpeechText = strGreeting

    Dim thread As System.Threading.Thread
    thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
    thread.Start()

    #####
End If

'Position Of Form...
Me.Size = New Size(2300, 1400)

End Sub

Private Function BuildLinkedDocsMenuList() As Boolean
    On Error GoTo Err_Fail

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

```

```

Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "Select * from tblLinkedDocs Order By Doc_Name"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "MyData")

Dim MaxRecCount As Integer
Dim inc As Integer
inc = -1
MaxRecCount = ds.Tables("MyData").Rows.Count

Dim menu As New MenuStrip()

For inc = 0 To MaxRecCount - 1
    'Dont Add Empty Options !!!
    Dim item As New ToolStripMenuItem(ds.Tables("MyData").Rows(inc).Item(0),
Nothing, AddressOf mnuLinkedDocItem_Clicked)
    item.Tag = ds.Tables("MyData").Rows(inc).Item(1)
    LinkedDocumentsToolStripMenuItem.DropDownItems.Add(item)
Next inc
con.Close()

BuildLinkedDocsMenuList = True
Exit Function
Err_Fail:
MsgBox(ErrorToString)
con.Close()
End Function

Private Sub mnuLinkedDocItem_Clicked(sender As Object, e As EventArgs)
    On Error GoTo Err_Fail

    'Process.Start(sender.Tag)
    Process.Start(strAppPath & "Linked Documents\" & sender.Tag)

Exit Sub
Err_Fail:
MsgBox("No Such Document Found", MsgBoxStyle.Exclamation, "Linked Document
Error")

MsgBox(strAppPath & "Linked Documents\" & sender.Tag)

End Sub

Private Sub SwitchCompanyToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles SwitchCompanyToolStripMenuItem.Click

Select Case strCompany
    Case "Tricera-Shop"
        strCompany = "Farout-Savings"
    Case "Farout-Savings"
        strCompany = "WitchOne.uk"
    Case "WitchOne.uk"

```

```

        strCompany = "Tricera-Shop"

    End Select

    'frmOutstanding.lblCompany.Text = strCompany

    'frmOutstanding.picCompany.Load("G:\Farout\" & strCompany & ".jpg")
    frmOutstanding.picCompany.Load(strAppPath & "\Resources\" & strCompany & ".jpg")

    MsgBox("Company Now Changed To '" & strCompany & "'", MsgBoxStyle.Information,
"Company")

    End Sub

    Private Function ShowReport(intWidth As Integer, intHeight As Integer, intX As
Integer, intY As Integer, strRpt As String) As Boolean
        On Error GoTo Err_Fail
        Dim cryRpt As New CrystalDecisions.CrystalReports.Engine.ReportDocument
        cryRpt.Load(strRpt)

        Dim ReportForm As New frmReports
        ReportForm.MdiParent = Me
        ReportForm.Show()
        ReportForm.Text = "Gross Sales Breakdown"
        ReportForm.Width = intWidth
        ReportForm.Height = intHeight
        ReportForm.Left = intX
        ReportForm.Top = intY

        ReportForm.rptLabelViewer.ReportSource = cryRpt
        ReportForm.rptLabelViewer.Refresh()

        ShowReport = True

    Exit Function
Err_Fail:
    MsgBox("Error Loading Requested Report", vbCritical, "Report Load Error")
    End Function

    Private Sub GrossSalesBreakdownToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles GrossSalesBreakdownToolStripMenuItem.Click

        'ShowReport(864, 715, 1032, 2, "G:\Farout\External
Reports\Gross_Sales_Breakdown.rpt")
        ShowReport(864, 700, 1000, 2, strAppPath & "External
Reports\Gross_Sales_Breakdown.rpt")

    End Sub

    Private Sub Last31DaysGrossToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles Last31DaysGrossToolStripMenuItem.Click

        'ShowReport(864, 715, 1032, 2, "G:\Farout\External
Reports\31day_Gross_Takings.rpt")
        ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\31day_Gross_Takings.rpt")

```

End Sub

```
Private Sub Last31DaysToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles Last31DaysToolStripMenuItem.Click
```

```
    'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\31day_Gross_ebay_Takings.rpt")  
    ShowReport(864, 700, 1032, 2, strAppPath & "External Reports\31day_Gross_ebay_Takings.rpt")
```

End Sub

```
Private Sub IncomeFromSeedsOrdersToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles IncomeFromSeedsOrdersToolStripMenuItem.Click
```

```
    'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\Seed_Sales_By_Month.rpt")  
    ShowReport(864, 700, 1032, 2, strAppPath & "External Reports\Seed_Sales_By_Month.rpt")
```

End Sub

```
Private Sub GrossCashSalesToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles GrossCashSalesToolStripMenuItem.Click
```

```
    'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\CASH_Takings.rpt")  
    ShowReport(864, 700, 1032, 2, strAppPath & "External Reports\CASH_Takings.rpt")
```

End Sub

```
Private Sub GrossEBaySalesToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles GrossEBaySalesToolStripMenuItem.Click
```

```
    'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\ebay_Takings.rpt")  
    ShowReport(864, 700, 1032, 2, strAppPath & "External Reports\ebay_Takings.rpt")
```

End Sub

```
Private Sub GrossBagSalesToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles GrossBagSalesToolStripMenuItem.Click
```

```
    'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\Bag_Takings.rpt")  
    ShowReport(864, 700, 1032, 2, strAppPath & "External Reports\Bag_Takings.rpt")
```

End Sub

```
Private Sub AnalysisToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles AnalysisToolStripMenuItem.Click
```

```
    'ShowReport(864, 983, 1032, 2, "G:\Farout\External Reports\Master_Orders_Breakdown.rpt")  
    ShowReport(864, 700, 1032, 2, strAppPath & "External Reports\Master_Orders_Breakdown.rpt")
```

End Sub



```

Private Sub Last31DaysEBayOrdersToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles Last31DaysEBayOrdersToolStripMenuItem.Click

    'ShowReport(864, 715, 1032, 2, "G:\Farout\External
Reports\Last_31_Days_eBay_Orders.rpt")
    ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\Last_31_Days_eBay_Orders.rpt")

End Sub

Private Sub SeedOrderBreakdownToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles SeedOrderBreakdownToolStripMenuItem.Click

    'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\Bonsai_Breakdown.rpt")
    ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\Bonsai_Breakdown.rpt")

End Sub

Private Sub PostageToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles PostageToolStripMenuItem.Click

    frmPostages.MdiParent = Me
    frmPostages.Show()

End Sub

Private Sub PostagesToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles PostagesToolStripMenuItem.Click

    frmPostages.MdiParent = Me
    frmPostages.Show()

End Sub

Private Sub LinkedDocumentsToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles LinkedDocumentsToolStripMenuItem.Click

End Sub

Private Sub AccountsToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles AccountsToolStripMenuItem.Click
    frmAccounts.MdiParent = Me
    frmAccounts.Show()
End Sub

Private Sub OpenSourceDirectoryToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles OpenSourceDirectoryToolStripMenuItem.Click
    On Error GoTo Err_Fail

    'Process.Start("G:\Farout\")
    Process.Start(strAppPath)

Exit Sub

```

```

Err_Fail:
    MsgBox("Could Not Open Data Source Directory.", MsgBoxStyle.Exclamation, "Linked
Directory Error.")
    End Sub

    Private Sub QUIToolStripMenuItem1_Click(sender As System.Object, e As
System.EventArgs) Handles QUIToolStripMenuItem1.Click

        frmSettings.MdiParent = Me
        frmSettings.Show()

    End Sub

    Private Sub PaypalCapitalAccountToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles PaypalCapitalAccountToolStripMenuItem.Click

        'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\Last356Day_Tricera-
Shop_Takings.rpt")
        ShowReport(864, 700, 1032, 2, strAppPath & "External Reports\Last356Day_Tricera-
Shop_Takings.rpt")

    End Sub

    Private Sub ExpensesByMonthToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ExpensesByMonthToolStripMenuItem.Click
        'ShowReport(864, 715, 1032, 2, "G:\Farout\External Reports\ExpensesByMonth.rpt")
        ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\ExpensesByMonth.rpt")
    End Sub

    Private Sub StockPurchasingByMonthToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
StockPurchasingByMonthToolStripMenuItem.Click
        'ShowReport(864, 715, 1032, 2, "G:\Farout\External
Reports\StockPurchasingByMonth.rpt")
        ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\StockPurchasingByMonth.rpt")
    End Sub

    Private Sub StockPurchasesBreakdownByTypeToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
StockPurchasesBreakdownByTypeToolStripMenuItem.Click
        'ShowReport(864, 715, 1032, 2, "G:\Farout\External
Reports\StockBreakdownByType.rpt")
        ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\StockBreakdownByType.rpt")
    End Sub

    Private Sub QuickTextToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles QuickTextToolStripMenuItem.Click
        frmQuickText.MdiParent = Me
        frmQuickText.Show()
    End Sub

    Private Sub MonthlyComparisonToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles MonthlyComparisonToolStripMenuItem.Click

```

```

Dim cryRpt As New CrystalDecisions.CrystalReports.Engine.ReportDocument

'cryRpt.Load("G:\Farout\External Reports\MonthlyIncomeComparison.rpt")
cryRpt.Load(strAppPath & "External Reports\MonthlyIncomeComparison.rpt")

'Either Use Current Month Or Ask User (Depending On Settings)
Dim strMonth As String
Dim currentDate As DateTime = DateTime.Now
strMonth = currentDate.Month

If blnUseCurrMonth = False Then
    strMonth = InputBox("Please Enter A Valid Month Number", "Date Rrequired",
strMonth)
End If

cryRpt.SetParameterValue("intMonth", strMonth)

'Get Month Name To Display On Report
Dim strDate As New DateTime(1, strMonth, 1)
strMonth = strDate.ToString("MMM")

'Display Compiled Report
Dim LabelFrm As New frmReports
LabelFrm.MdiParent = Me
LabelFrm.Show()
LabelFrm.Text = "(ebay) Monthly Comparison Report For (" & strMonth & ")"

LabelFrm.Width = 864
LabelFrm.Height = 700
LabelFrm.Left = 1032
LabelFrm.Top = 2

LabelFrm.rptLabelViewer.ReportSource = cryRpt
LabelFrm.rptLabelViewer.Refresh()

End Sub

Private Sub PaypalSalesByCategoryToolStripMenuItem_Click(sender As System.Object, e
As System.EventArgs) Handles PaypalSalesByCategoryToolStripMenuItem.Click

    'ShowReport(864, 715, 1032, 2, "G:\Farout\External
Reports\PaypalSalesByCategory.rpt")
    ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\PaypalSalesByCategory.rpt")

End Sub

Private Sub DrawingsForSelfToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles DrawingsForSelfToolStripMenuItem.Click
    'ShowReport(864, 715, 1032, 2, "G:\Farout\External
Reports\Last4Weeks_DrawingsForSelf.rpt")
    ShowReport(864, 700, 1032, 2, strAppPath & "External
Reports\Last4Weeks_DrawingsForSelf.rpt")

```

```
End Sub
End Class
```

```
Imports System.Net.Mail
```

```
Public Class frmOrders
```

```
Private Function SearchOptions(strOption As String) As Boolean
    On Error GoTo Err_Fail
```

```
    'Clear List Contents
    Me.cboQSList.Items.Clear()
    Me.cboQSList.Text = ""
```

```
    Select Case strOption
```

```
        Case "Customers"
```

```
            If chkSearchFilter.CheckState = 1 Then
                ' Display Customers Without Outstanding Orders (Filter Option By
```

```
Default)
```

```
                Call FillDataList(Me.cboQSList, "Select distinct
tblCustomers.CustomerCode & ' : ' & tblCustomers.name from tblCustomers INNER JOIN
tblOrders ON tblCustomers.CustomerCode = tblOrders.CustomerCode WHERE tblOrders.Paid Is
Null")
```

```
            Else
```

```
                ' Display All Customers (No Filter)
```

```
                Call FillDataList(Me.cboQSList, "Select CustomerCode & ' : ' & Name
from tblCustomers Order By CustomerCode DESC")
```

```
            End If
```

```
        Case "Stock"
```

```
            ' Top 15 Items Sold
```

```
            Call FillDataList(Me.cboQSList, "SELECT top 15
tblOrders.SalesDescription, sum(tblOrders.qty) AS [Count], tblOrders.ItemCode From
tblOrders WHERE (((tblOrders.ItemCode) <> 'P99999999') AND
((tblOrders.OriginalDate)>Now()-31)) GROUP BY tblOrders.SalesDescription,
tblOrders.ItemCode ORDER BY sum(tblOrders.qty) DESC")
```

```
            'Call FillDataList(Me.cboQSList, "Select StockCode & ' : ' & ShortDesc
from tblStock Order By StockCode DESC")
```

```
        Case "Ebay Customers"
```

```
            Call FillDataList(Me.cboQSList, "Select top 20 OrderNum & ' : ' &
ebayname from tblDeliveryAdds where len(ebayname)>0 Order By OrderNum DESC")
```

```
    End Select
```

```
    SearchOptions = True
    Exit Function
```

```
Err_Fail:
```

```
    SearchOptions = False
```

End Function

```
Private Sub optCustomers_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles optCustomers.CheckedChanged
```

```
Me.chkSearchFilter.Visible = True  
Me.chkSearchFilter.Enabled = True  
Me.chkSearchFilter.Text = "Filter"
```

```
Me.cboQSList.Width = 172
```

```
Me.cboOutstanding.Enabled = True  
Me.cboOutstanding.Items.Clear()  
Me.cboOutstanding.Text = ""
```

```
Call SearchOptions("Customers")
```

```
Me.lstResults.Items.Clear()
```

End Sub

```
Private Sub optStock_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles optStock.CheckedChanged
```

```
Me.chkSearchFilter.Visible = False  
Me.chkSearchFilter.Enabled = False  
Me.cboQSList.Width = 234  
Me.cboOutstanding.Enabled = False  
Me.cboOutstanding.Items.Clear()  
Me.cboOutstanding.Text = ""  
Call SearchOptions("Stock")  
Me.lstResults.Items.Clear()
```

End Sub

```
Private Sub optEbayCustomers_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles optEbayCustomers.CheckedChanged
```

```
Me.chkSearchFilter.Visible = False  
Me.chkSearchFilter.Enabled = False  
Me.cboQSList.Width = 234  
Me.cboOutstanding.Enabled = False  
Me.cboOutstanding.Items.Clear()  
Me.cboOutstanding.Text = ""  
Call SearchOptions("Ebay Customers")  
Me.lstResults.Items.Clear()
```

End Sub

```
Private Sub chkSearchFilter_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles chkSearchFilter.CheckedChanged
```

```
Me.chkSearchFilter.Visible = True
```

```
Me.chkSearchFilter.Enabled = True
Me.chkSearchFilter.Text = "Filter"
Me.cboQSList.Width = 172
'Me.cboOutstanding.Enabled = False
Me.cboOutstanding.Items.Clear()
Me.cboOutstanding.Text = ""
Call SearchOptions("Customers")
```

End Sub

```
Private Sub btnSearch_Click(sender As System.Object, e As System.EventArgs) Handles
btnSearch.Click
```

```
txtSearch.Text = Trim(txtSearch.Text)
```

```
'Clear Contents Of Results Table
Me.lstResults.Items.Clear()
```

```
If Len(txtSearch.Text) < 1 Then Exit Sub
```

```
Dim strSQL As String = ""
```

```
If optCustomers.Checked = True Then
    strSQL = "Select CustomerCode & ' : ' & Name From tblCustomers where Name
LIKE '%" & txtSearch.Text & "%' ORDER BY [CustomerCode] DESC"
End If
```

```
If optStock.Checked = True Then
    strSQL = "Select StockCode & ' : ' & Shortdesc From tblStock where ShortDesc
LIKE '%" & txtSearch.Text & "%' ORDER BY [ShortDesc] ASC"
End If
```

```
If optEbayCustomers.Checked = True Then
    strSQL = "Select OrderNum & ' : ' & ebayname from tblDeliveryAdds where
ebayname LIKE '%" & txtSearch.Text & "%' Order By OrderNum DESC"
End If
```

```
Call FillDataList(Me.lstResults, strSQL)
```

End Sub

```
Public Sub cboQSList_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboQSList.SelectedIndexChanged
    On Error GoTo ERR_Fail
```

```
If optCustomers.Checked = True Then
```

```
    Dim strSQL As String
```

```

        Dim strCustCode As String
        strCustCode = Mid(Me.cboQSLList.Text, 1, 7)
        strSQL = "SELECT DISTINCT tblOrders.OrderNum, tblorders.CustomerCode FROM
tblorders WHERE isnull(tblOrders.paid) and tblorders.CustomerCode = '" & strCustCode & "'
Order By tblOrders.OrderNum DESC"
        Call FillDataList(Me.cboOutstanding, strSQL)

        Me.lblCustomerCode.Text = strCustCode

    End If

    If optStock.Checked = True Then
        Dim strStockC As String
        ' Clear Stock Form Data
        Call ClearOrdersFrmData()
        strStockC = GetStockCodeFromDesc(cboQSLList.Text)
        If GetStockItem(strStockC) = False Then
            MsgBox("Error Loading Stock Item Data", MsgBoxStyle.Critical)
            ' Clear Stock Form Data
            Call ClearOrdersFrmData()
        End If
    End If

    If optEbayCustomers.Checked = True Then
        'Load Selected Order !
        Dim strCode As String
        strCode = Mid(Me.cboQSLList.Text, 1, 11)

        If LoadExistingOrder(strCode) = False Then
            MsgBox("Error Loading Existing Order Data", MsgBoxStyle.Critical)
            'Call ClearStockFrmData()
        End If

    End If

    'Build List Of Most Popular Items For This Customer
    Call BuildPopularItemsMenu()

    Exit Sub
ERR_Fail:
    MsgBox(ErrorToString)
End Sub

```

```

Private Function GetStockItem(strStockCode As String) As Boolean
    On Error GoTo ERR_Fail

```

```

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

```

```

Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
sql = "select * from tblStock Where StockCode = '" & strStockCode & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "tblStock")

'.ToString Protects Against Null/Empty
Me.txtShortDescription.Text =
ds.Tables("tblStock").Rows(0).Item("ShortDesc").ToString
Me.txtDescription.Text =
ds.Tables("tblStock").Rows(0).Item("Description").ToString
Me.txtPrice.Text = ds.Tables("tblStock").Rows(0).Item("Price").ToString
Me.txtEbayID.Text = ds.Tables("tblStock").Rows(0).Item("EbayItemCode").ToString
Me.txtCost.Text = ds.Tables("tblStock").Rows(0).Item("Cost").ToString
Me.lblStockCode.Text = ds.Tables("tblStock").Rows(0).Item("StockCode").ToString
Me.cboQty.Text = "1"

con.Close()

GetStockItem = True
Exit Function
ERR_Fail:
GetStockItem = False
MsgBox(ErrorToString)
End Function

Private Function ClearOrdersFrmData() As Boolean
On Error GoTo Err_Fail

'Stock Form Tab
Me.txtShortDescription.Text = ""
Me.txtDescription.Text = ""
Me.txtPrice.Text = "0"
Me.txtEbayID.Text = ""
Me.txtCost.Text = "0"
Me.lblStockCode.Text = ""
Me.cboQty.Text = ""

'Address Details Form Tab
Me.cboTitle.Text = ""
Me.txtFirstName.Text = ""
Me.txtSurname.Text = ""
Me.txtAddr1.Text = ""
Me.txtAddr2.Text = ""
Me.txtAddr3.Text = ""
Me.txtTown.Text = ""
Me.txtPCode.Text = ""
Me.txtAddressNotes.Text = ""
Me.txtEmail.Text = ""
Me.txtEbayUserID.Text = ""
Me.cboPostMethod.Text = ""
Me.txtPostDate.Text = ""
Me.txtTrackingRef.Text = ""
Me.cboPayMethod.Text = ""
Me.txtReported.Text = ""
Me.txtInvoiceMessage.Text = ""
Me.txtReceived.Text = "0.00"

```



```

Me.chkSmartStamp.Checked = False
Me.chkFlagged.Checked = False
Me.chkPacked.Checked = False

Me.chkLostInPost.Checked = False
Me.chkCancelled.Checked = False

Me.cboPostMethod.Text = ""
Me.cboPayMethod.Text = ""

Me.lblOrderValueTotal.Text = "£00.00"
Me.lblProfit.Text = "(£0.00)"

ClearOrdersFrmData = True
Exit Function
Err_Fail:
ClearOrdersFrmData = False
End Function

Private Sub frmOrders_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

'Setup Order Grid Sizes & Columns
dgOrderGrid.ColumnCount = 8
Dim column As DataGridViewColumn = dgOrderGrid.Columns(1)
column.MinimumWidth = 30
'DISPLAYED COLUMNS
dgOrderGrid.Columns(0).Name = "Description"
dgOrderGrid.Columns(0).Width = 185
dgOrderGrid.Columns(1).Name = "Qty"
dgOrderGrid.Columns(1).Width = 30
dgOrderGrid.Columns(2).Name = "Price"
dgOrderGrid.Columns(2).Width = 60
dgOrderGrid.Columns(3).Name = "Total"
dgOrderGrid.Columns(3).Width = 63
'HIDDEN COLUMNS
dgOrderGrid.Columns(4).Name = "ItemCode"
dgOrderGrid.Columns(5).Name = "Cost"
dgOrderGrid.Columns(6).Name = "CustCode"
dgOrderGrid.Columns(7).Name = "eBayID"

'Set Order Date Label
Me.lblOrderDate.Text = Format(Now(), "dd/MM/yyyy")

'Fill Combo Choices From NEW Database Table

Call FillDataList(Me.cboPostMethod, "Select PostMethod from tblListBoxOptions")
Call FillDataList(Me.cboPayMethod, "Select PayMethod from tblListBoxOptions")
Call FillDataList(Me.cboTitle, "Select Titles from tblListBoxOptions")
'Call FillDataList(Me.cboPostDate, "Select PostDates from tblListBoxOptions")

'Alternate Row Colour
dgOrderGrid.DefaultCellStyle.BackColor = Color.White
dgOrderGrid.AlternatingRowsDefaultCellStyle.BackColor = Color.Cornsilk
'Color.AliceBlue

'#####

```

```

Call Build_MultiUseMenu()

Call BuildPostDateMenu()

'Load strCompany Logo
'Me.picCompany.Load("G:\Farout\" & strCompany & ".jpg")
Me.picCompany.Load(strAppPath & "\Resources\" & strCompany & ".jpg")

```

```
End Sub
```

```
Private Function BuildPopularItemsMenu() As Boolean
    On Error GoTo Err_Fail
```

```

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

```

```

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String

```

```

    'sql = "SELECT top 10 tblOrders.SalesDescription, sum(tblOrders.qty) AS [Count],
tblOrders.ItemCode From tblOrders WHERE (((tblOrders.ItemCode) <> 'P99999999') AND
((tblOrders.OriginalDate)>Now()-90) AND (tblorders.customercode = '' &
Me.lblCustomerCode.Text & '')) GROUP BY tblOrders.SalesDescription,
tblOrders.ItemCode ORDER BY sum(tblOrders.qty) DESC"

```

```

'For New Count Method (Ignoring Variations) Call Function To Convert ItemCode To
Description

```

```

    sql = "SELECT top 20 tblOrders.ItemCode as ITEMCODES, sum(tblOrders.qty) AS
[Count], tblOrders.ItemCode From tblOrders WHERE (((tblOrders.ItemCode) <> 'P99999999')
AND ((tblOrders.OriginalDate)>Now()-90) AND (tblorders.customercode = '' &
Me.lblCustomerCode.Text & '')) GROUP BY tblOrders.ItemCode ORDER BY
sum(tblOrders.qty) DESC"

```

```

    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "tblStock")
    Dim MaxRecCount As Integer
    Dim inc As Integer
    Dim strListItem As String
    inc = -1

```

```
MaxRecCount = ds.Tables("tblStock").Rows.Count
```

```

'DESTROY EXISTING MENU ITEMS FIRST
cmDynamicMenu.Items.Clear()

```

```

    Dim strMenuItemText As String
    Dim strMenuItemName As String
    Me.txtShortDescription.ContextMenuStrip = cmDynamicMenu

```

```

Dim strShortDesc As String

For inc = 0 To MaxRecCount - 1

'#####
####
' For New Count Method (Ignoring Variations) Call Function To Convert
ItemCode To Description

'GetSHORTStockItemDescription
strShortDesc =
GetSHORTStockItemDescription(ds.Tables("tblStock").Rows(inc).Item("ItemCode"))

'#####
####

' strMenuItemText = "(" & ds.Tables("tblStock").Rows(inc).Item("Count") & ") "
& ds.Tables("tblStock").Rows(inc).Item("Salesdescription")

strMenuItemText = "(" & ds.Tables("tblStock").Rows(inc).Item("Count") & ") "
& strShortDesc

strMenuItemName = ds.Tables("tblStock").Rows(inc).Item("ItemCode")

'MsgBox(strMenuItemName)

'Work out how to use inc in "cmDynamicMenu.Items.Add(menu0)" !!!

Select Case inc
    Case 0
        Dim menu0 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu0.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu0)
    Case 1
        Dim menu1 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu1.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu1)
    Case 2
        Dim menu2 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu2.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu2)
    Case 3
        Dim menu3 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu3.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu3)

```

```

    Case 4
        Dim menu4 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu4.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu4)
    Case 5
        Dim menu5 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu5.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu5)
    Case 6
        Dim menu6 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu6.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu6)
    Case 7
        Dim menu7 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu7.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu7)
    Case 8
        Dim menu8 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu8.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu8)
    Case 9
        Dim menu9 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu9.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu9)
    Case 10
        Dim menu10 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu10.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu10)

    Case 11
        Dim menu11 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu11.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu11)

    Case 12
        Dim menu12 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu12.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu12)

    Case 13
        Dim menu13 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu13.Click, AddressOf mnuItem_Clicked
        cmDynamicMenu.Items.Add(menu13)

    Case 14
        Dim menu14 As New ToolStripMenuItem() With {.Text = strMenuItemText,
.Name = strMenuItemName}
        AddHandler menu14.Click, AddressOf mnuItem_Clicked

```

```

        cmDynamicMenu.Items.Add(menu14)

    End Select

Next

con.Close()

BuildPopularItemsMenu = True
Exit Function
Err_Fail:
    MsgBox(ErrorToString)
End Function

Private Sub mnuItem_Clicked(sender As Object, e As EventArgs)
    cmDynamicMenu.Hide() 'Sometimes the menu items can remain open. May not be
    necessary for you.

    'Identify Control Clicked...Workround for Mnu item opening on everything!
    If cmDynamicMenu.SourceControl.Name.ToString = "txtShortDescription" Or
    cmDynamicMenu.SourceControl.Name.ToString = "picPopularItems" Then

        Dim item As ToolStripMenuItem = TryCast(sender, ToolStripMenuItem)
        If item IsNot Nothing Then
            GetStockItem(item.Name)
        End If
    End If

End Sub

Private Sub btnADD_Click(sender As System.Object, e As System.EventArgs) Handles
    btnADD.Click

    'Clear Data Binding From Order Grid ???

    'Validate Required Data
    If Len(txtShortDescription.Text) < 1 Then Exit Sub
    If Len(lblCustomerCode.Text) <> 7 Then Exit Sub
    If Mid(lblCustomerCode.Text, 1, 1) <> "C" Then Exit Sub
    If Len(Me.lblOrderNumber.Text) <> 11 Then Exit Sub
    If Mid(Me.lblOrderNumber.Text, 1, 3) <> "ORD" Then Exit Sub

    'Remove Currency Sign (If Any)
    Me.txtCost.Text = WithCurrencySign(Me.txtCost.Text, False)
    Me.txtPrice.Text = WithCurrencySign(Me.txtPrice.Text, False)

```

```
    Dim myRow As String() = New String() {Me.txtShortDescription.Text,  
Me.cboQty.Text, FormatCurrency(Me.txtPrice.Text, True),  
FormatCurrency(Val(Me.cboQty.Text) * Val(Me.txtPrice.Text), True), Me.lblStockCode.Text,  
FormatCurrency(Val(Me.txtCost.Text), True), Me.lblCustomerCode.Text, Me.txtEbayID.Text}
```

```
    dgOrderGrid.Rows.Add(myRow)
```

```
    'Calculate Grid Total
```

```
    lblOrderValueTotal.Text = FormatCurrency(GetGridTotal(Me.dgOrderGrid, 3), -1)
```

```
    'Calculate Profit
```

```
    Me.lblProfit.Text = "(" & FormatCurrency(GetGridPROFIT(), -1) & ")"
```

```
End Sub
```

```
Private Function GetGridPROFIT() As Double
```

```
    On Error GoTo Err_Fail
```

```
    Dim X As Integer
```

```
    GetGridPROFIT = 0
```

```
    Dim currLineProfit As Double
```

```
    Dim intQty As Integer
```

```
    Dim currTotal As Double
```

```
    Dim currCost As Double
```

```
    'Loop Through Datagrid
```

```
    For X = 0 To dgOrderGrid.RowCount - 1
```

```
        'Protect From NULL Values In Grid
```

```
        If (dgOrderGrid.Rows(X).Cells(1).Value) IsNot DBNull.Value Then
```

```
            intQty = dgOrderGrid.Rows(X).Cells(1).Value
```

```
            currTotal = dgOrderGrid.Rows(X).Cells(3).Value
```

```
            currCost = dgOrderGrid.Rows(X).Cells(5).Value
```

```
            currLineProfit = currTotal - (intQty * currCost)
```

```
            If currLineProfit <> 0 Then
```

```
                GetGridPROFIT = GetGridPROFIT + currLineProfit
```

```
            End If
```

```
        End If
```

```
    Next X
```

```
    Exit Function
```

```
Err_Fail:
```

```
    GetGridPROFIT = 0
```

```
    MsgBox(ErrorToString)
```

```
End Function
```

```
Private Sub dgOrderGrid_Click(sender As Object, e As System.EventArgs) Handles dgOrderGrid.Click
```

```
    Dim i As Integer
    i = Me.dgOrderGrid.CurrentRow.Index

    'Get Data From Datagrid
    Me.txtShortDescription.Text = dgOrderGrid.Item(0, i).Value
    Me.cboQty.Text = dgOrderGrid.Item(1, i).Value
    'Take Currency Sign back Off
    Me.txtPrice.Text = WithCurrencySign(dgOrderGrid.Item(2, i).Value, False)
    Me.txtEbayID.Text = dgOrderGrid.Item(7, i).Value
    Me.lblStockCode.Text = dgOrderGrid.Item(4, i).Value
    Me.txtCost.Text = dgOrderGrid.Item(5, i).Value
    'Get Long Description From Stock Table (But NOT For Postage Items)
    Me.txtDescription.Text = ""
    Me.txtDescription.Text = GetStockItemDescription(Me.lblStockCode.Text)

    'Remove Selected Row From Datagrid
    Me.dgOrderGrid.Rows.RemoveAt(i)

    'Calculate Grid Total
    lblOrderValueTotal.Text = FormatCurrency(GetGridTotal(Me.dgOrderGrid, 3), -1)

    'Calculate Profit
    Me.lblProfit.Text = "(" & FormatCurrency(GetGridPROFIT(), -1) & ")"

End Sub
```

```
Private Sub btnUKPost_Click(sender As System.Object, e As System.EventArgs) Handles btnUKPost.Click
```

```
    ' Add Standard UK Postage Details
    Me.txtPrice.Text = GetPostageCharge(Me.lblStockCode.Text, True)
    Me.txtShortDescription.Text = "Postage & Packing"
    Me.txtDescription.Text = ""
    Me.lblStockCode.Text = "P99999999"
    Me.cboQty.Text = "1"
    Me.txtEbayID.Text = ""
    Me.txtCost.Text = "0.00"
```

```
End Sub
```

```
Private Sub btnAirmail_Click(sender As System.Object, e As System.EventArgs) Handles btnAirmail.Click
```

```
    ' Add Airmail Postage Details
    Me.txtPrice.Text = GetPostageCharge(Me.lblStockCode.Text, False)
    Me.txtShortDescription.Text = "Postage & Packing"
    Me.txtDescription.Text = ""
    Me.lblStockCode.Text = "P99999999"
    Me.cboQty.Text = "1"
    Me.txtEbayID.Text = ""
    Me.txtCost.Text = "0.00"
```

End Sub

```
Private Sub btnCreateNewOrder_Click(sender As System.Object, e As System.EventArgs)
Handles btnCreateNewOrder.Click
    'On Error GoTo Err_Fail

    'Check That A Customer Has Been Selected
    Dim strCustomerCode As String = Me.lblCustomerCode.Text
    Dim blnValidated As Boolean = True
    'Validation Checks
    If Len(strCustomerCode) <> 7 Then blnValidated = False
    If Mid(strCustomerCode, 1, 1) <> "C" Then blnValidated = False
    'Result ?
    If blnValidated = False Then
        MsgBox("You Have Not Selected A Valid Customer", MsgBoxStyle.Information)
        Exit Sub
    End If

    'Check Date Required, Today Or Other Day
    If Me.chkCreateNewTODAY.CheckState = 1 Then

        Me.lblOrderDate.Text = Now().ToString("dd/MM/yyyy")

    Else
        Dim strDate As String
        strDate = InputBox("Please Enter A Valid Date", "Order Date",
Date.Today.AddDays(-1))
        If IsDate(strDate) = False Then
            MsgBox("Invalid Date...", MsgBoxStyle.Exclamation, "Date Error")
            Exit Sub
        End If
        lblOrderDate.Text = strDate
    End If

    '#####

    ' Create & Display A New Order Number

    '#####
    Me.lblOrderNumber.Text = CreateNewORDERNumber()

    If blnSPEECH = True Then

        'Send Speech To Multithread Process
        strMTSpeechText = GetSPEECH("NewOrderCreated", "Anytime", "n")
        Dim thread As System.Threading.Thread
        thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
        thread.Start()

    Else
```



```
MsgBox("A New Order Number " & Me.lblOrderNumber.Text & ", Has Now Been  
Generated...", MsgBoxStyle.Information)
```

```
End If
```

```
'#####
```

```
'Clear Any Existing Form Data  
Call ClearOrdersFrmData()  
Me.dgOrderGrid.Rows.Clear()
```

```
BuildPopularItemsMenu()  
Me.btnSaveOrder.Enabled = True
```

```
Me.lblOrderNumber.ForeColor = Color.Navy  
Me.lblEMail.ForeColor = Color.Black
```

```
'#####
```

```
'Me.picCompany.Load("G:\Farout\" & frmOutstanding.lblCompany.Text & ".jpg")
```

```
'Me.picCompany.Load("G:\Farout\" & strCompany & ".jpg")  
Me.picCompany.Load(strAppPath & "\Resources\" & strCompany & ".jpg")
```

```
'frmOutstanding.lblCompany.Text
```

```
'#####
```

```
Exit Sub  
'Err_Fail:  
'MsgBox(ErrorToString)  
End Sub
```

```
Private Sub lstResults_DoubleClick(sender As Object, e As System.EventArgs) Handles  
lstResults.DoubleClick  
On Error GoTo Err_Fail
```

```
Dim strCode As String = ""
```

```
If optCustomers.Checked = True Then
```

```
strCode = Mid(Me.lstResults.SelectedItem.ToString(), 1, 7)  
Me.cboOutstanding.Items.Clear()  
Dim strSQL As String  
strSQL = "SELECT DISTINCT tblOrders.OrderNum, tblOrders.CustomerCode FROM  
tblOrders WHERE isnull(tblOrders.paid) and tblOrders.CustomerCode = '" & strCode & "'  
Order By tblOrders.OrderNum DESC"  
Call FillDataList(Me.cboOutstanding, strSQL)
```

```

        Me.lblCustomerCode.Text = strCode

    End If

    If optStock.Checked = True Then
        strCode = Mid(Me.lstResults.SelectedItem.ToString(), 1, 9)

        ' Clear JUST Stock Item Form Data
        Me.txtShortDescription.Text = ""
        Me.txtDescription.Text = ""
        Me.txtPrice.Text = ""
        Me.txtEbayID.Text = ""
        Me.txtCost.Text = ""
        Me.lblStockCode.Text = ""
        Me.cboQty.Text = ""

        If GetStockItem(strCode) = False Then
            MsgBox("Error Loading Stock Item Data", MsgBoxStyle.Critical)
            ' Clear JUST Stock Item Form Data
            Me.txtShortDescription.Text = ""
            Me.txtDescription.Text = ""
            Me.txtPrice.Text = ""
            Me.txtEbayID.Text = ""
            Me.txtCost.Text = ""
            Me.lblStockCode.Text = ""
            Me.cboQty.Text = ""

        End If

    End If

    If optEbayCustomers.Checked = True Then
        'Load Selected Order !
        strCode = Mid(Me.lstResults.SelectedItem.ToString(), 1, 11)

        If LoadExistingOrder(strCode) = False Then
            MsgBox("Error Loading Existing Order Data", MsgBoxStyle.Critical)
            'Call ClearStockFrmData()
        End If

    End If

    'MsgBox(strCode)
    Exit Sub
Err_Fail:

    End Sub

    Private Sub btnClose1_Click(sender As System.Object, e As System.EventArgs) Handles
    btnClose1.Click

        Me.Dispose()

    End Sub

```

```

Public Function LoadExistingOrder(strOrderNumber As String) As Boolean
    On Error GoTo Err_Fail

    Me.lblOrderNumber.ForeColor = Color.Navy
    Me.lblEmail.ForeColor = Color.Black

    'Clear All Existing Order Form Data & Release Datagrid Of Any Previous Data Binds
    Call ClearOrdersFrmData()
    Me.dgOrderGrid.Rows.Clear()
    'Me.dgOrderGrid.DataMember = Nothing
    'Me.dgOrderGrid.DataSource = Nothing

    'Update Order Number
    Me.lblOrderNumber.Text = strOrderNumber

    'Update Customer Code
    Me.lblCustomerCode.Text = GetCustomerCodeFromOrder(strOrderNumber)

    'Update Original Order Date
    Me.lblOrderDate.Text = GetOriginalOrderDate(strOrderNumber)

    'Set Tabstrip To First Tab
    Me.tabCurrentOrder.SelectTab(0)

    'Allow Selection Of New Stock Items From Popular Items List
    Me.optCustomers.Checked = True
    Call BuildPopularItemsMenu()

    'Build SQL Recordset
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()

    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String

    sql = "SELECT tblOrders.SalesDescription, tblOrders.Qty,
format(tblOrders.Price,'currency') as Price, format((tblOrders.Qty *
tblOrders.Price),'currency') as Total, tblOrders.ItemCode, tblOrders.Cost,
tblOrders.CustomerCode, tblOrders.Notes, tblOrders.eBayID from tblOrders WHERE
tblOrders.OrderNum='" & strOrderNumber & "'"

    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "MyData")
    Dim MaxRecCount As Integer
    Dim inc As Integer
    inc = -1
    MaxRecCount = ds.Tables("MyData").Rows.Count

    Dim strShortDescription As String
    Dim strQTY As String
    Dim dblPrice As Double
    Dim dblTotal As Double

```

```

Dim dblCost As Double
Dim strStockCode As String
Dim strCustCode As String
Dim strEbayID As String
For inc = 0 To MaxRecCount - 1
    strShortDescription = ds.Tables("MyData").Rows(inc).Item("SalesDescription")
    strQTY = ds.Tables("MyData").Rows(inc).Item("Qty")
    dblPrice = ds.Tables("MyData").Rows(inc).Item("Price")
    dblTotal = (ds.Tables("MyData").Rows(inc).Item("Price")) * (Val(strQTY))
    strStockCode = ds.Tables("MyData").Rows(inc).Item("ItemCode")
    dblCost = ds.Tables("MyData").Rows(inc).Item("Cost")
    strCustCode = ds.Tables("MyData").Rows(inc).Item("CustomerCode")
    strEbayID = ds.Tables("MyData").Rows(inc).Item("eBayID")

    Dim myRow As String() = New String() {strShortDescription, strQTY,
FormatCurrency(dblPrice, True), FormatCurrency(dblTotal, True), strStockCode, dblCost,
strCustCode, strEbayID}
    dgOrderGrid.Rows.Add(myRow)
Next

con.Close()

'Calculate Grid Total
lblOrderValueTotal.Text = FormatCurrency(GetGridTotal(Me.dgOrderGrid, 3), -1)

'Calculate Profit
Me.lblProfit.Text = "(" & FormatCurrency(GetGridPROFIT(), -1) & ")"

'Check If Order Is Already Complete
If IsOrderComplete(strOrderNumber) = True Then
    btnSaveOrder.Enabled = False
    dgOrderGrid.Enabled = False

    Me.grpBoxCOMPLETE.Enabled = False
Else
    btnSaveOrder.Enabled = True
    dgOrderGrid.Enabled = True
    Me.grpBoxCOMPLETE.Enabled = True
End If

'Set Global COMPANY Setting To Suit Order Company (On Both Outstanding & Order
Forms)
Dim sOrder As String = GetOrderCompany(strOrderNumber)
'frmOutstanding.lblCompany.Text = sOrder

'frmOutstanding.picCompany.Load("G:\Farout\" & sOrder & ".jpg")
frmOutstanding.picCompany.Load(strAppPath & "\Resources\" & sOrder & ".jpg")

'Me.picCompany.Load("G:\Farout\" & sOrder & ".jpg")
Me.picCompany.Load(strAppPath & "\Resources\" & sOrder & ".jpg")

strCompany = sOrder

If LoadDeliveryDetails(strOrderNumber) = False Then
    LoadExistingOrder = False
    Exit Function
End If

```

```

        LoadExistingOrder = True
    Exit Function
Err_Fail:
    LoadExistingOrder = False
    MsgBox(ErrorToString)
End Function

Private Sub cboOutstanding_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cboOutstanding.SelectedIndexChanged

    'Load Selected Order !
    Dim strCode As String
    strCode = Trim(Me.cboOutstanding.Text)

    If LoadExistingOrder(strCode) = False Then
        MsgBox("Error Loading Existing Order Data", MsgBoxStyle.Critical)
        'Call ClearStockFrmData()
    End If

End Sub

Public Function LoadDeliveryDetails(strOrderNumber As String) As Boolean
    On Error Resume Next

    'Build SQL Recordset
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "SELECT * From tblDeliveryAdds WHERE tblDeliveryAdds.OrderNum='" &
strOrderNumber & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "MyData")

    Me.cboTitle.Text = ds.Tables("MyData").Rows(0).Item("Title")
    Me.txtFirstName.Text = ds.Tables("MyData").Rows(0).Item("FirstName")
    Me.txtSurname.Text = ds.Tables("MyData").Rows(0).Item("Surname")
    Me.txtAddr1.Text = ds.Tables("MyData").Rows(0).Item("Addr1")
    Me.txtAddr2.Text = ds.Tables("MyData").Rows(0).Item("Addr2")
    Me.txtAddr3.Text = ds.Tables("MyData").Rows(0).Item("Addr3")
    Me.txtTown.Text = ds.Tables("MyData").Rows(0).Item("Town")
    Me.txtPCode.Text = ds.Tables("MyData").Rows(0).Item("PCode")
    Me.txtAddressNotes.Text = ds.Tables("MyData").Rows(0).Item("Notes")
    Me.txtEmail.Text = ds.Tables("MyData").Rows(0).Item("Email")

```

```

'Extra Data For Email Communications
Me.txtSubject.Text = "Your Recent Order With " &
GetOrderCompany(Me.lblOrderNumber.Text)

If InStr(GetOrderCompany(Me.lblOrderNumber.Text), "Tricera") > 0 Then
    Me.txtFrom.Text = "sales@tricera-shop.co.uk"
Else
    Me.txtFrom.Text = "sales@witchone.uk"
End If

Me.txtEbayUserID.Text = ds.Tables("MyData").Rows(0).Item("eBayName")
Me.cboPostMethod.Text = ds.Tables("MyData").Rows(0).Item("Post_Method")

Me.txtPostDate.Text = ds.Tables("MyData").Rows(0).Item("DispatchDate")

If Me.txtPostDate.Text = "01/01/1999" Then Me.txtPostDate.Text = ""

Me.txtTrackingRef.Text = ds.Tables("MyData").Rows(0).Item("TrackingRef")
Me.cboPayMethod.Text = ds.Tables("MyData").Rows(0).Item("PayMethod")

Me.txtReported.Text = ds.Tables("MyData").Rows(0).Item("UnpaidItemDate")

If Me.txtReported.Text = "01/01/1999" Then Me.txtReported.Text = ""

Me.txtInvoiceMessage.Text = ds.Tables("MyData").Rows(0).Item("InvoiceMessage")
Me.txtReceived.Text = "£" & ds.Tables("MyData").Rows(0).Item("OrderValue")

Me.chkSmartStamp.Checked = ds.Tables("MyData").Rows(0).Item("SmartStamp")
Me.chkFlagged.Checked = ds.Tables("MyData").Rows(0).Item("Flag")
Me.chkPacked.Checked = ds.Tables("MyData").Rows(0).Item("AlreadyPacked")

'#####

Me.chkLostInPost.Checked = ds.Tables("MyData").Rows(0).Item("LostInPost")
Me.chkCancelled.Checked = ds.Tables("MyData").Rows(0).Item("Cancelled")

'#####

Me.txtTo.Text = ds.Tables("MyData").Rows(0).Item("Email")

con.Close()

'Set Global COMPANY Setting To Suit Order Company

LoadDeliveryDetails = True
Exit Function

```

```

Err_Fail:
    LoadDeliveryDetails = False
    MsgBox(ErrorToString)
End Function

Private Sub btnIMPORT_Click(sender As System.Object, e As System.EventArgs) Handles
btnIMPORT.Click
    On Error GoTo Err_Fail
    'Don't Allow Empty Search Text
    If Len(Me.txtEbayUserID.Text) = 0 Then Exit Sub
    'Build SQL Recordset
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String
    sql = "SELECT * From tblDeliveryAdds WHERE tblDeliveryAdds.eBayName='" &
Trim(Me.txtEbayUserID.Text) & "'"
    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "MyData")
    'Protect Against Nothing Found
    If ds.Tables("MyData").Rows.Count > 0 Then
        Me.cboTitle.Text = ds.Tables("MyData").Rows(0).Item("Title")
        Me.txtFirstName.Text = ds.Tables("MyData").Rows(0).Item("FirstName")
        Me.txtSurname.Text = ds.Tables("MyData").Rows(0).Item("Surname")
        Me.txtAddr1.Text = ds.Tables("MyData").Rows(0).Item("Addr1")
        Me.txtAddr2.Text = ds.Tables("MyData").Rows(0).Item("Addr2")
        Me.txtAddr3.Text = ds.Tables("MyData").Rows(0).Item("Addr3")
        Me.txtTown.Text = ds.Tables("MyData").Rows(0).Item("Town")
        Me.txtPCode.Text = ds.Tables("MyData").Rows(0).Item("PCode")
        Me.txtEmail.Text = ds.Tables("MyData").Rows(0).Item("Email")
    End If

    con.Close()
Exit Sub
Err_Fail:
    MsgBox(ErrorToString)
End Sub

Private Function BuildEmail() As Boolean
    On Error GoTo Err_Fail

    Dim strGreeting As String
    If Len(Me.txtEbayUserID.Text) > 0 Then
        strGreeting = "Dear " & Me.txtEbayUserID.Text & ", " & Chr(13) & Chr(13)
    Else
        strGreeting = "Dear " & Me.cboTitle.Text & " " & Me.txtFirstName.Text & " "
& Me.txtSurname.Text & ", " & Chr(13) & Chr(13)
    End If

    'eBay Managed Payment Introduced...(All Payments Made Through Ebay)

```

```

'strGreeting = strGreeting & "Many Thanks For Your Purchase And " &
Me.cboPayMethod.Text & " Payment." & Chr(13) & Chr(13)

strGreeting = strGreeting & "Many Thanks For Your Purchase." & Chr(13) & Chr(13)

Dim strDispatchDetails As String
Dim mydate As DateTime
If IsDate(Me.txtPostDate.Text) = True Then
    mydate = DateTime.Parse(Me.txtPostDate.Text)
Else
    Me.rtfEmail.Text = Chr(13) & Chr(13) & "Required Data Missing...(Post Date)"
    BuildEmail = False
    Exit Function
End If
strDispatchDetails = "Your order will be dispatched by " & Me.cboPostMethod.Text
& " Post, On " & Format(mydate, "dddd dd MMMM , yyyy") & "." & Chr(13) & Chr(13)

'#####

'Does Order Have A Tracking Number? (BEFORE DISPATCH)
If Len(Me.txtTrackingRef.Text) > 0 Then
    Dim strTrackingDetails As String

    Select Case Len(txtTrackingRef.Text)

        Case 13
            strTrackingDetails = "Your Order Tracking Number Is '" &
Me.txtTrackingRef.Text & "', Which Can Be Tracked Here ; http://www.royalmail.com/" &
Chr(13) & Chr(13)
        Case 11
            strTrackingDetails = "Your Order Tracking Number Is '" &
Me.txtTrackingRef.Text & "', Which Can Be Tracked Here ; https://www.parcel2go.com/" &
Chr(13) & Chr(13)
        Case 16
            strTrackingDetails = "Your Order Tracking Number Is '" &
Me.txtTrackingRef.Text & "', Which Can Be Tracked Here ; https://www.myhermes.co.uk/" &
Chr(13) & Chr(13)
        Case Else
            strTrackingDetails = ""
    End Select

    strDispatchDetails = strDispatchDetails & strTrackingDetails

End If

'#####

Dim strAddressDetails As String
strAddressDetails = "We Will Dispatch Your Order To The Following Address;" &
Chr(13) & Chr(13)
strAddressDetails = strAddressDetails & Me.cboTitle.Text & " " &
Me.txtFirstName.Text & " " & Me.txtSurname.Text & Chr(10)
strAddressDetails = strAddressDetails & Me.txtAddr1.Text & Chr(10)
strAddressDetails = strAddressDetails & Me.txtAddr2.Text & Chr(10)

```



```

strAddressDetails = strAddressDetails & Me.txtAddr3.Text & Chr(10)
strAddressDetails = strAddressDetails & Me.txtTown.Text & Chr(10)
strAddressDetails = strAddressDetails & Me.txtPCode.Text & Chr(13) & Chr(13)

If Len(strAddressDetails) < (64 + 20) Then

    Me.rtfEmail.Text = Chr(13) & Chr(13) & "Required Data Missing...(Address
Details)"
    BuildEmail = False
    Exit Function

End If

Dim strOrderDetails As String
strOrderDetails = "Order Details;" & Chr(13) & Chr(13)
Dim X As Integer
Dim currValue As Double
'Read Order Items From Datagrid
For X = 0 To dgOrderGrid.RowCount - 1
    'Protect From NULL Values In Grid
    If (dgOrderGrid.Rows(X).Cells(1).Value) IsNot DBNull.Value Then
        'Dont Include Blank Rows
        If Len(dgOrderGrid.Rows(X).Cells(1).Value) > 0 Then
            strOrderDetails = strOrderDetails & "Qty : " & dgOrderGrid.Item(1,
X).Value & "    " & dgOrderGrid.Item(0, X).Value & Chr(10)
        End If
    End If
Next X
strOrderDetails = strOrderDetails & Chr(13)

Dim strClosingComment As String
strClosingComment = "When your item arrives, please take the time to leave
feedback, and we will do the same for you." & Chr(13) & Chr(13)
strClosingComment = strClosingComment & "If there are any problems with your
order, please contact us before leaving feedback, to give us the opportunity to help." &
Chr(13) & Chr(13)

Dim strKindRegards As String

'Get Order Company

Dim strTemp As String = GetOrderCompany(Me.lblOrderNumber.Text)

Select Case strTemp
    Case "Farout-Savings"
        strKindRegards = "Kind Regards," & Chr(10) &
GetOrderCompany(Me.lblOrderNumber.Text) & Chr(13) & Chr(13)
    Case "Tricera-Shop"
        strKindRegards = "Kind Regards," & Chr(10) & "www." &
GetOrderCompany(Me.lblOrderNumber.Text) & ".co.uk" & Chr(13) & Chr(13)
    Case "WitchOne.uk"
        strKindRegards = "Kind Regards," & Chr(10) & "www." &
GetOrderCompany(Me.lblOrderNumber.Text) & Chr(13) & Chr(13)
    Case Else
        strKindRegards = "Kind Regards," & Chr(10)

```

```

End Select

'strKindRegards = "Kind Regards," & Chr(10) & "www." &
GetOrderCompany(Me.lblOrderNumber.Text) & ".co.uk" & Chr(13) & Chr(13)

'Display eMail Details Onto Form
Me.rtfEmail.Text = strGreeting & strDispatchDetails & strAddressDetails &
strOrderDetails & strClosingComment & strKindRegards

BuildEmail = True
Exit Function
Err_Fail:
BuildEmail = False
'Clear All Email Details...

MsgBox(ErrorToString)
End Function

Private Sub tabCurrentOrder_Click(sender As Object, e As System.EventArgs) Handles
tabCurrentOrder.Click

'Decide If Order Is Ready For Email Or Not
'Complete, Paid, Etc

'Check What Tab Has Been Clicked
If tabCurrentOrder.SelectedTab.Name = "tabCommunications" Then

'Do We Need An Email?
Select Case Me.cboPayMethod.Text
Case "INVOICED", "UNPAID", "Cash"
Me.rtfEmail.Text = ""
Case Else

'Decide Which Company Details To Use
Select Case strCompany

Case "Tricera-Shop"
Me.txtSubject.Text = "Your eBay Order With " & strCompany
Me.txtFrom.Text = "sales@tricera-shop.co.uk"

Case Else
Me.txtSubject.Text = "Your eBay Order With " & strCompany
Me.txtFrom.Text = "sales@witchone.uk"

End Select

Me.txtTo.Text = Me.txtEmail.Text

```

```

        Call BuildEmail()

    End Select

End If

End Sub

Private Sub btnPasteAddress_Click(sender As System.Object, e As System.EventArgs)
Handles btnPasteAddress.Click

    On Error Resume Next 'GoTo Err_Fail

    Dim strFullAddress As String
    Dim intX As Integer : Dim intX2 As Integer : Dim xStart As Integer : Dim XEnd As
Integer

    'Check Clipboard For TEXT (ONLY) Address
    Dim iData As IDataObject = Clipboard.GetDataObject()
    If iData.GetDataPresent(DataFormats.Text) Then
        'Clipboard Contains Text So Get It And Continue...
        strFullAddress = CType(iData.GetData(DataFormats.Text), String)
    Else
        'Clipboard Doesn't Contain Text So Exit
        Exit Sub
    End If

    '#####
    'Check For eMail Text...
    If InStr(1, strFullAddress, "@", 1) > 0 Then

        MsgBox("'" & strFullAddress & "' is an email address..?",
MsgBoxStyle.Exclamation)

        If Len(Trim(txtEmail.Text)) = 0 Then

            txtEmail.Text = (Trim(strFullAddress))

        End If

        'Send Speech To Multithread Process
        strMTSpeechText = "Clipboard Data Contains An Email Address"
        Dim thread As System.Threading.Thread
        thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
        thread.Start()
    End If
End Sub

```

```

        Exit Sub
    End If
    '#####

Dim AddressArray(10) As String : Dim intArray As Integer
'Get First Name (Upto First Space)
For X = 1 To Len(strFullAddress)

    If Mid(strFullAddress, X, 1) = " " Then
        AddressArray(1) = (Mid(strFullAddress, 1, X - 1))
        xStart = X
        Exit For
    End If
Next
'Get Second Name (From First Space To End Of Line Char)
For X = xStart To Len(strFullAddress)
    'Find END Of Surname
    If Asc(Mid(strFullAddress, X, 1)) = 9 Or Asc(Mid(strFullAddress, X, 1)) = 13
Then
        AddressArray(2) = Mid(strFullAddress, xStart + 1, X - xStart).ToString
        xStart = X + 1
        Exit For
    End If
Next

'Treat Address Commas As New Line...
Dim strRetChar As String = Environment.NewLine
strFullAddress = strFullAddress.Replace(",", strRetChar)

'Loop Through Each Line Of Rest Of Address
For intArray = 3 To 7
    For X = xStart To Len(strFullAddress)
        'Find END Of First Line
        If Asc(Mid(strFullAddress, X, 1)) = 9 Or Asc(Mid(strFullAddress, X, 1)) =
13 Then
            AddressArray(intArray) = Mid(strFullAddress, xStart + 1, X -
xStart).ToString
            xStart = X + 1
            Exit For
        End If
    Next
Next

AddressArray(0) = ""
Me.cboTitle.Text = AddressArray(0)

'Write Array Contents To Form & Trim

Me.txtFirstName.Text = AddressArray(1).Trim()

```

```

Me.txtSurname.Text = AddressArray(2).Trim()
Me.txtAddr1.Text = AddressArray(3).Trim()
Me.txtAddr2.Text = AddressArray(4).Trim()
Me.txtAddr3.Text = AddressArray(5).Trim()
Me.txtTown.Text = AddressArray(6).Trim()
Me.txtPCode.Text = AddressArray(7).Trim()

Exit Sub
'Err_Fail:
' MsgBox(ErrorToString)
End Sub

```

```

Private Sub btnProperCase_Click(sender As System.Object, e As System.EventArgs)
Handles btnProperCase.Click

```

```

'Convert Each Line To Propercase
Me.txtFirstName.Text = StrConv(Me.txtFirstName.Text, vbProperCase)
Me.txtSurname.Text = StrConv(Me.txtSurname.Text, vbProperCase)
Me.txtAddr1.Text = StrConv(Me.txtAddr1.Text, vbProperCase)
Me.txtAddr2.Text = StrConv(Me.txtAddr2.Text, vbProperCase)
Me.txtAddr3.Text = StrConv(Me.txtAddr3.Text, vbProperCase)
Me.txtTown.Text = StrConv(Me.txtTown.Text, vbProperCase)
Me.txtPCode.Text = StrConv(Me.txtPCode.Text, vbProperCase)

'Find and Format PostCodes
If IsPostCode(Me.txtPCode.Text) = True Then Me.txtPCode.Text =
FormatPostCode(Me.txtPCode.Text)
If IsPostCode(Me.txtTown.Text) = True Then Me.txtTown.Text =
FormatPostCode(Me.txtTown.Text)
If IsPostCode(Me.txtAddr3.Text) = True Then Me.txtAddr3.Text =
FormatPostCode(Me.txtAddr3.Text)
If IsPostCode(Me.txtAddr2.Text) = True Then Me.txtAddr2.Text =
FormatPostCode(Me.txtAddr2.Text)

'Guess Title
Me.cboTitle.Text = GetSuggestedTitle(Me.txtFirstName.Text)

```

```

' Look For Special Case Names (McDonalds)

```

```

End Sub

```

```

Private Sub btnSaveOrder_Click(sender As System.Object, e As System.EventArgs)
Handles btnSaveOrder.Click

```

```

If dgOrderGrid.RowCount = 0 Then
    MsgBox("There are no items to save !", MsgBoxStyle.Exclamation, "No Items")

```

```

Exit Sub

End If

'Make Required Declarations...
Dim X As Integer
Dim sql As String
Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
Dim upd As New OleDb.OleDbCommand
Dim strOrderNum, strItemCode, streBayID, strCustomerCode, strSalesDescription As
String
Dim currCost, currPrice As Double
Dim intQty As Integer
Dim dteOriginalDate As Date

strOrderNum = Me.lblOrderNumber.Text.Trim
strCustomerCode = Me.lblCustomerCode.Text.Trim

'#####
'#####

'NO ITEMS NO SAVE - Otherwise tbldelivery adds gets an order record, without
matching tblorders !!

'#####
'#####

'Decide If Order Is NEW or EXISTING
If ReturnRecordCount("Select * from tblOrders Where OrderNum = '" &
lblOrderNumber.Text.Trim & "'") = 0 Then
    If MsgBox("Do You Wish To Create New Order Record?", MsgBoxStyle.YesNo +
MsgBoxStyle.Question, "") = vbNo Then Exit Sub
    '### NEW ORDER ###
    'Create An Order Entry Into tblDeliveryAdds
    sql = "Insert into tblDeliveryAdds ([OrderNum]) values (?)" ',[DispatchDate]
    con.Open()
    Try
        upd = New OleDb.OleDbCommand(sql, con)
        'sql(Parameter)
        upd.Parameters.Add(New OleDb.OleDbParameter("OrderNum", strOrderNum))
        'upd.Parameters.Add(New OleDb.OleDbParameter("DispatchDate",
"01/01/1999"))

        upd.ExecuteNonQuery()
        con.Close()

```

```

        Catch ex As Exception
            MsgBox.Show("SAVE FAILED (Manual Data Validation Required!) : " &
                Err.Description, "Caution")
        End Try

    Else
        If MsgBox("Do You Wish To Overwrite Existing Order?", MsgBoxStyle.YesNo +
            MsgBoxStyle.Question, "") = vbNo Then Exit Sub
        '### EXISTING ORDER ###
        'Delete Existing Order Items From Database (Ready To Save New Items)
        sql = "Delete * From tblOrders where OrderNum = '" & strOrderNum & "'"
        con.Open()
        upd = New OleDb.OleDbCommand(sql, con)
        upd.ExecuteNonQuery()
        con.Close()
    End If

    'Loop Through Datagrid & Save Order Items To tblOrders (Required For Saving NEW
    And Existing Orders)

    For X = 0 To dgOrderGrid.RowCount - 1
        'Protect From NULL Values In Grid
        If (dgOrderGrid.Rows(X).Cells(0).Value) IsNot DBNull.Value Then
            con.Open()
            'strOrderNum = Me.lblOrderNumber.Text.Trim
            strItemCode = dgOrderGrid.Rows(X).Cells(4).Value.ToString
            streBayID = dgOrderGrid.Rows(X).Cells(7).Value.ToString
            currCost = dgOrderGrid.Rows(X).Cells(5).Value
            currPrice = dgOrderGrid.Rows(X).Cells(2).Value
            intQty = dgOrderGrid.Rows(X).Cells(1).Value
            'strCustomerCode = Me.lblCustomerCode.Text.Trim

            dteOriginalDate = Me.lblOrderDate.Text & "Now().ToString("dd/MM/yyyy")

            strSalesDescription = dgOrderGrid.Rows(X).Cells(0).Value

            sql = "Insert into tblOrders
                ([OrderNum],[ItemCode],[eBayID],[Cost],[Price],[Qty],[CustomerCode],[OriginalDate],[Sales
                Description],[Company] ) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
            Try
                upd = New OleDb.OleDbCommand(sql, con)
                'SQL Parameter List...
                upd.Parameters.Add(New OleDb.OleDbParameter("OrderNum", strOrderNum))
                upd.Parameters.Add(New OleDb.OleDbParameter("ItemCode", strItemCode))
                upd.Parameters.Add(New OleDb.OleDbParameter("eBayID", streBayID))
                upd.Parameters.Add(New OleDb.OleDbParameter("Cost", currCost))
                upd.Parameters.Add(New OleDb.OleDbParameter("Price", currPrice))
                upd.Parameters.Add(New OleDb.OleDbParameter("Qty", intQty))
                upd.Parameters.Add(New OleDb.OleDbParameter("CustomerCode",
strCustomerCode))
                upd.Parameters.Add(New OleDb.OleDbParameter("OriginalDate",
dteOriginalDate))
                upd.Parameters.Add(New OleDb.OleDbParameter("SalesDescription",
strSalesDescription))
                upd.Parameters.Add(New OleDb.OleDbParameter("Company", strCompany))
            Catch ex As Exception
                MsgBox.Show("SAVE FAILED (Manual Data Validation Required!) : " &
                    Err.Description, "Caution")
            End Try
        End If
    Next X

```

```

        upd.ExecuteNonQuery()
        con.Close()
        upd.Dispose()
    Catch ex As Exception
        MsgBox.Show("SAVE FAILED (Manual Data Validation Required!) : " &
Err.Description, "Caution")
        con.Close()
    End Try
End If
Next X

'#####

If blnSPEECH = True Then

    'Check strCompany Against tblImpediment (For Words That Speech Engine Cant
Say :-)
    Dim strAlternativeSpelling As String = GetAlternativeSpelling(strCompany)

    'Send Speech To Multithread Process
    strMTSpeechText = GetSPEECH("SaveSuccess", "Anytime", "n") & " To Company " &
strAlternativeSpelling
    Dim thread As System.Threading.Thread
    thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
    thread.Start()

Else
    MsgBox("Order Saved Successfully To Company : " & strCompany & ".",
MsgBoxStyle.Information, "Saved")
End If

'#####

End Sub

Private Sub btnClose2_Click(sender As System.Object, e As System.EventArgs) Handles
btnClose2.Click
    Me.Dispose()
End Sub

Private Sub btnAddrDetailsSAVE_Click(sender As System.Object, e As System.EventArgs)
Handles btnAddrDetailsSAVE.Click

    'Make Required Declarations...
    Dim sql As String
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    Dim upd As New OleDb.OleDbCommand
    Dim strOrderNum As String = Me.lblOrderNumber.Text.Trim
    con.Open()

```



```

        sql = "UPDATE tblDeliveryAdds SET [Title]=@Title, [FirstName]=@FirstName ,
[Surname]=@Surname, [Addr1]=@Addr1, [Addr2]=@Addr2, [Addr3]=@Addr3, [Town]=@Town,
[Pcode]=@Pcode,"
        sql = sql & " [Notes]=@Notes, [PayMethod]=@PayMethod , [eBayName]=@eBayName,
[eMail]=@eMail, [OrderValue]=@OrderValue,"
        If Len(Me.txtPostDate.Text) > 0 Then
            sql = sql & " [DispatchDate]=@DispatchDate,"
        End If
        sql = sql & " [Post_Method]=@Post_Method, [InvoiceMessage]=@InvoiceMessage,"
        If Len(Me.txtReported.Text) > 0 Then
            sql = sql & " [UnpaidItemDate]=@UnpaidItemDate,"
        End If
        sql = sql & " [TrackingRef]=@TrackingRef, [SmartStamp]=@SmartStamp, [Flag]=@Flag,
[AlreadyPacked]=@AlreadyPacked,"

'#####
sql = sql & " [LostInPost]=@LostInPost, [Cancelled]=@Cancelled"
'#####

sql = sql & " WHERE OrderNum = '" & strOrderNum & "'"

Try
    upd = New OleDb.OleDbCommand(sql, con)
    'SQL Parameter List...
    upd.Parameters.Add(New OleDb.OleDbParameter("Title", Me.cboTitle.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("FirstName",
Me.txtFirstName.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("Surname", Me.txtSurname.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("Addr1", Me.txtAddr1.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("Addr2", Me.txtAddr2.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("Addr3", Me.txtAddr3.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("Town", Me.txtTown.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("Pcode", Me.txtPCode.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("Notes",
Me.txtAddressNotes.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("PayMethod",
Me.cboPayMethod.Text))
    upd.Parameters.Add(New OleDb.OleDbParameter("eBayName",
Trim(Me.txtEbayUserID.Text)))
    upd.Parameters.Add(New OleDb.OleDbParameter("eMail", Trim(Me.txtEmail.Text)))

    ' upd.Parameters.Add(New OleDb.OleDbParameter("OrderValue",
Me.txtReceived.Text))

    upd.Parameters.Add(New OleDb.OleDbParameter("OrderValue",
WithCurrencySign(Me.txtReceived.Text, False)))

    If Len(Me.txtPostDate.Text) > 0 Then
        upd.Parameters.Add(New OleDb.OleDbParameter("DispatchDate",
Me.txtPostDate.Text))
    End If
    upd.Parameters.Add(New OleDb.OleDbParameter("Post_Method",
Me.cboPostMethod.Text))

```

```

        upd.Parameters.Add(New OleDb.OleDbParameter("InvoiceMessage",
Me.txtInvoiceMessage.Text))
        If Len(Me.txtReported.Text) > 0 Then
            upd.Parameters.Add(New OleDb.OleDbParameter("UnpaidItemDate",
Me.txtReported.Text))
        End If
        upd.Parameters.Add(New OleDb.OleDbParameter("TrackingRef",
Trim(Me.txtTrackingRef.Text)))
        Dim bln As Boolean
        bln = Me.chkSmartStamp.Checked
        upd.Parameters.AddWithValue("SmartStamp", bln)
        bln = Me.chkFlagged.Checked
        upd.Parameters.AddWithValue("Flag", bln)

        bln = Me.chkPacked.Checked
        upd.Parameters.AddWithValue("AlreadyPacked", bln)

        '#####
        bln = Me.chkLostInPost.Checked
        upd.Parameters.AddWithValue("LostInPost", bln)

        bln = Me.chkCancelled.Checked
        upd.Parameters.AddWithValue("Cancelled", bln)

        '#####

        upd.ExecuteNonQuery()
        con.Close()
        upd.Dispose()

    If blnSPEECH = True Then
        'Dim SAPI
        'SAPI = CreateObject("SAPI.spvoice")
        'SAPI.Speak(GetSPEECH("SaveSuccess", "Anytime", "n"))

        'Send Speech To Multithread Process
        strMTSpeechText = GetSPEECH("SaveSuccess", "Anytime", "n")
        Dim thread As System.Threading.Thread
        thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
        thread.Start()

    Else
        MsgBox("Delivery Details Saved Successfully", MsgBoxStyle.Information,
"Saved")
    End If

    Catch ex As Exception
        MsgBox.Show("SAVE FAILED (Manual Data Validation Required!) : " &
Err.Description, "Caution")
        con.Close()
        upd.Dispose()
    End Try

End Sub

```

```

Private Sub btnSendEmail_Click(sender As System.Object, e As System.EventArgs)
Handles btnSendEmail.Click

    'Validate Details
    Dim strMSG As String

    'Send Email
    If SendOutlookMail(Me.txtFrom.Text, Me.txtTo.Text, Me.rtfEmail.Text) = False Then
        'MsgBox("Error Sending Email...", MsgBoxStyle.Critical, "Email")
        strMTSpeechText = GetSPEECH("CommsFail", "Anytime", "n")
        strMSG = "Error Sending Email."
    Else
        'MsgBox("Email Sent Successfully", MsgBoxStyle.Information, "Email Sent...")
        strMTSpeechText = GetSPEECH("CommsSuccess", "Anytime", "n")
        strMSG = "Email Sent Successfully."
    End If

    '#####

    If blnSPEECH = True Then

        Dim thread As System.Threading.Thread
        thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
        thread.Start()

    Else
        Select Case strMSG
            Case "Email Sent Successfully."
                MsgBox(strMSG, MsgBoxStyle.Information, "Saved")
            Case "Error Sending Email."
                MsgBox(strMSG, MsgBoxStyle.Exclamation, "ERROR...")
            Case Else

        End Select

    End If

    '#####

    Exit Sub

End Sub

Public Function SendOutlookMail(txtFrom As String, txtTo As String, txtMessage As
String) As Boolean

    On Error GoTo Err_Fail
    Dim oMailapp As Object
    Dim oMailItem As Object

```

```

oMailapp = CreateObject("Outlook.application")
oMailItem = oMailapp.createitem(0)

'Get Outlook Account Index To Match Company
Dim strCompanyEmail As String = ""
Dim intOutlookAccountIndex As Integer

Select Case strCompany
    Case "Tricera-Shop"
        strCompanyEmail = "sales@tricera-shop.co.uk"
    Case "Farout-Savings"
        strCompanyEmail = "sales@farout-savings.co.uk"
    Case "WitchOne.uk"
        strCompanyEmail = "sales@WitchOne.uk"
End Select

intOutlookAccountIndex = GetOutlookEmailAccountIndex(oMailapp,
strCompanyEmail.ToLower)

'Send Email To Outlook
'
With oMailItem
    .Subject = "Your Recent Order With " & strCompany
    .To = txtTo
    .body = txtMessage
    .Importance = 2
    .SendUsingAccount = oMailItem.session.Accounts.Item(intOutlookAccountIndex)
    .Send()
    .Display()
End With

'
oMailapp = Nothing
oMailItem = Nothing
'

SendOutlookMail = True
Exit Function
Err_Fail:
MsgBox(ErrorToString)
oMailapp = Nothing
oMailItem = Nothing
SendOutlookMail = False
End Function

Function GetOutlookEmailAccountIndex(objOutlook As Object, strEmail As String) As
Integer
    GetOutlookEmailAccountIndex = 0
    Dim X As Integer = 1
    For Each objOAccount In objOutlook.Session.Accounts
        If objOAccount.DisplayName = strEmail Then
            GetOutlookEmailAccountIndex = X
            Exit Function
        End If
        X = X + 1
    Next
End Function

```

```

    Next objOAccount
End Function

Public Function SendMail(txtFrom As String, txtTo As String, txtMessage As String) As Boolean

    Try
        Dim Sntp_Server As New SntpClient
        Dim e_mail As New MailMessage()
        Sntp_Server.UseDefaultCredentials = False
        'Decide Which Company Details To Use
        Select Case strCompany
            Case "Tricera-Shop"
                Sntp_Server.Credentials = New Net.NetworkCredential("sales@tricera-shop.co.uk", "trikemonster")
                e_mail.Subject = "Your eBay Order With Tricera-Shop"
            Case "Farout-Savings"
                Sntp_Server.Credentials = New Net.NetworkCredential("sales@farout-savings.co.uk", "foraptor79")
                e_mail.Subject = "Your eBay Order With Farout-Savings"
        End Select
        Sntp_Server.Port = 587
        Sntp_Server.EnableSsl = True
        Sntp_Server.Host = "auth.smtp.1and1.co.uk"
        e_mail = New MailMessage()
        e_mail.From = New MailAddress(txtFrom)
        e_mail.To.Add(txtTo)
        e_mail.Subject = Me.txtSubject.Text
        e_mail.IsBodyHtml = False
        e_mail.Body = txtMessage

        'Code works in IDE but Fails in compiled application => Timing ?

        Threading.Thread.Sleep(500)

        Sntp_Server.Send(e_mail)

        Threading.Thread.Sleep(500)

        SendMail = True
    Exit Function

    Catch error_t As Exception
        SendMail = False
        MsgBox(error_t.ToString)
    End Try

End Function

Private Function BuildCOMPLETEDemail() As String
    On Error GoTo Err_Fail

    Dim strMSGBody As String
    Dim strDispatchDte As String

    If Len(Me.txtEbayUserID.Text) > 0 Then

```

```

        strMSGBody = "Dear " & Me.txtEbayUserID.Text & ", " & Chr(13) & Chr(13)
    Else
        strMSGBody = "Dear " & Me.cboTitle.Text & " " & Me.txtFirstName.Text & " " &
Me.txtSurname.Text & ", " & Chr(13) & Chr(13)
    End If

    Dim mydate As DateTime
    If IsDate(Me.txtPostDate.Text) = True Then
        mydate = DateTime.Parse(Me.txtPostDate.Text)
    Else
        MsgBox("Dispatch Date Format Error", MsgBoxStyle.Critical, "ERROR")
        BuildCOMPLETEDEmail = ""
        Exit Function
    End If

    strMSGBody = strMSGBody & "Your recent order was dispatched on " & Format(mydate,
"dd MMMM, yyyy")

    strMSGBody = strMSGBody & ", and should be with you shortly." & Chr(13) & Chr(13)

    strMSGBody = strMSGBody & "If you purchased your order through ebay then please
take the time to leave feedback, and we will do the same for you." & Chr(13) & Chr(13) &
"If there are any problems with your order, please contact us so that we can help." &
Chr(13) & Chr(13)

    strMSGBody = strMSGBody & "Your Unique Order Reference Number Is " &
Me.lblOrderNumber.Text & Chr(13) & Chr(13)

    Dim strOrderDetails As String = ""
    Dim X As Integer
    Dim currValue As Double
    'Read Order Items From Datagrid
    For X = 0 To dgOrderGrid.RowCount - 1
        'Protect From NULL Values In Grid
        If (dgOrderGrid.Rows(X).Cells(1).Value) IsNot DBNull.Value Then
            'Dont Include Blank Rows
            If Len(dgOrderGrid.Rows(X).Cells(1).Value) > 0 Then
                strOrderDetails = strOrderDetails & "Qty : " & dgOrderGrid.Item(1,
X).Value & " " & dgOrderGrid.Item(0, X).Value & Chr(10)
            End If
        End If
    Next X
    strMSGBody = strMSGBody & strOrderDetails & Chr(13) & Chr(13)

    'Does Order Have A Tracking Number?
    Dim strTrackingDetails As String

    If Len(txtTrackingRef.Text) > 0 Then

        strTrackingDetails = "Your Order Tracking Number Is '" & txtTrackingRef.Text
& "', And Can Be Tracked Here : " & Chr(13) & Chr(13)

        Select Case Len(txtTrackingRef.Text)
            Case 13
                'Royal Mail

```

```

        strTrackingDetails = strTrackingDetails &
"http://www.royalmail.com/portal/rm/track?trackNumber=" & Trim(Me.txtTrackingRef.Text)
        Case 11
            'Parcel2Go
            strTrackingDetails = strTrackingDetails &
"http://www.parcel2go.com/"
        Case 16
            'MyHermes
            strTrackingDetails = strTrackingDetails &
"http://www.myhermes.co.uk/"
        Case Else

    End Select

    strTrackingDetails = strTrackingDetails & Chr(13) & Chr(13)

    strMSGBody = strMSGBody & strTrackingDetails

Else
    strTrackingDetails = ""

End If

strMSGBody = strMSGBody & "Once again, thank you for your purchase." & Chr(13) &
Chr(13)

'Get Order Company

Dim strTemp As String = GetOrderCompany(Me.lblOrderNumber.Text)

Select Case strTemp
    Case "Farout-Savings"
        strMSGBody = strMSGBody & "Kind Regards," & Chr(10) & strTemp & Chr(13) &
Chr(13)
    Case "Tricera-Shop"
        strMSGBody = strMSGBody & "Kind Regards," & Chr(10) & "www." & strTemp &
".co.uk" & Chr(13) & Chr(13)
    Case "WitchOne.uk"
        'strMSGBody = strMSGBody & "Kind Regards," & Chr(10) & strTemp & Chr(13)
& Chr(13)
        strMSGBody = strMSGBody & "Kind Regards," & Chr(10) & "www." & strTemp &
".uk" & Chr(13) & Chr(13)
    Case Else
        strMSGBody = strMSGBody & "Kind Regards." & Chr(10)
End Select

'strMSGBody = strMSGBody & "Kind Regards," & Chr(10) & "www." &
GetOrderCompany(Me.lblOrderNumber.Text) & ".co.uk" & Chr(13) & Chr(13)

'#####CORONAVIRUS DELAYS#####

strMSGBody = strMSGBody & Chr(13) & Chr(13) & "**** PS. Please bear in mind that
due to the recent Coronavirus outbreak, some postal delays may occur. ****" & Chr(13)

```

```

        BuildCOMPLETEDEmail = strMSGBody

    Exit Function
Err_Fail:
    BuildCOMPLETEDEmail = ""
    MsgBox(ErrorToString)
End Function

Private Sub btnCompleteOrder_Click(sender As System.Object, e As System.EventArgs)
Handles btnCompleteOrder.Click

    'Save Post Date to [Paid] Field - Effectively Completing Order!
    Dim sql As String
    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim upd As New OleDb.OleDbCommand
    Dim strOrderNum As String = Me.lblOrderNumber.Text.Trim

'#####
    'Close Order...

    sql = "UPDATE tblOrders SET [Paid]= @Paid WHERE OrderNum = '" & strOrderNum & "'"

    Dim intCompleted As Integer = 0

    Try
        upd = New OleDb.OleDbCommand(sql, con)
        upd.Parameters.Add(New OleDb.OleDbParameter("Paid", Me.txtPostDate.Text))
        upd.ExecuteNonQuery()
        con.Close()

        intCompleted = intCompleted + 1 'Order Closed Successfully

'#####
        'Allow Completion Of Orders that Don't Have An Email Address

        If Len(Me.txtEmail.Text) > 3 Then
            'Send Completed Email But Only If Record Updates Successfully!
            If txtDispatchEmail.Checked = True Then

                If SendOutlookMail(Me.txtFrom.Text, Me.txtTo.Text,
BuildCOMPLETEDEmail) = False Then
                    'MsgBox("Error Sending Email...", MsgBoxStyle.Critical, "Email")
                Else
                    intCompleted = intCompleted + 2 ' Email Sent Successfully

                End If
            End If
        End If
    End Try
End Sub

```



```

                'MsgBox("Email Sent Successfully", MsgBoxStyle.Information, "Email
Sent...")

                End If
            End If

'#####

        Me.grpBoxCOMPLETE.Enabled = False

        'Has Everything Completed Successfully?

        Select Case intCompleted

            Case 1
                'strMTSpeechText = "This Order Has Now Been Closed"
                strMTSpeechText = GetSPEECH("OrderClosed", "Anytime", "n")
            Case 3
                'strMTSpeechText = "This Order Has Now Been Closed, And The Buyer Has
Been Notified"
                strMTSpeechText = GetSPEECH("OrderClosed", "Anytime", "n") & " And "
& GetSPEECH("CommsSuccess", "Anytime", "n")
            Case Else
                'strMTSpeechText = "There was an error completimg this order, please
check your data"
                strMTSpeechText = GetSPEECH("ErrorUnspecified", "Anytime", "n")
        End Select

        If blnSPEECH = True Then
            'Dim SAPI
            'SAPI = CreateObject("SAPI.spvoice")
            'SAPI.Speak(sql)

            'Send Speech To Multithread Process
            Dim thread As System.Threading.Thread
            thread = New System.Threading.Thread(AddressOf MultiThreadSpeech)
            thread.Start()
        Else
            MsgBox(strMTSpeechText, MsgBoxStyle.Information, "")
        End If

        Catch ex As Exception
            con.Close()
            MessageBox.Show("FAILED To Save Completed Date (No Email Sent!): " &
Err.Description, "Caution")
        End Try

        upd.Dispose()

```

```
End Sub
```

```
Private Sub btnEBay_Click(sender As System.Object, e As System.EventArgs) Handles  
btnEBay.Click
```

```
Dim strItem As String
```

```
If Len(txtEbayID.Text) = 12 And IsNumeric(txtEbayID.Text) Then
```

```
strItem = "http://cgi.ebay.co.uk/ws/eBayISAPI.dll?ViewItem&item=" &  
txtEbayID.Text & "&sspa"
```

```
Process.Start(strItem)
```

```
Else
```

```
strItem = "http://www.ebay.co.uk/"
```

```
Process.Start(strItem)
```

```
End If
```

```
End Sub
```

```
Private Sub btnTrack_Click(sender As System.Object, e As System.EventArgs) Handles  
btnTrack.Click
```

```
Dim strItem As String
```

```
Select Case Len(txtTrackingRef.Text)
```

```
Case 13
```

```
'Royal Mail
```

```
strItem = "http://www.royalmail.com/portal/rm/track?trackNumber=" &  
Trim(Me.txtTrackingRef.Text)
```

```
Case 11
```

```
'Parcel2Go
```

```
strItem = "https://www.parcel2go.com/"
```

```
Case 16
```

```
'MyHermes
```

```
strItem = "https://www.myhermes.co.uk/?crmLogout=true"
```

```
Case Else
```

```
Exit Sub
```

```
End Select
```

```
Process.Start(strItem)
```

```
End Sub
```

```
Private Sub btnCopyToClipboard_Click(sender As System.Object, e As System.EventArgs)  
Handles btnCopyToClipboard.Click
```

```
Clipboard.SetText(Me.rtfEmail.Text)
```

```
End Sub
```

```
Private Sub lblOrderNumber_Click(sender As System.Object, e As System.EventArgs)  
Handles lblOrderNumber.Click
```

```
Clipboard.SetText(Me.lblOrderNumber.Text)
```

```
Me.lblOrderNumber.ForeColor= Color.Green
```

```
End Sub
```

```

Private Sub cboPayMethod_SelectedValueChanged(sender As Object, e As
System.EventArgs) Handles cboPayMethod.SelectedValueChanged

    Select Case Me.cboPayMethod.Text

        Case "INVOICED", "UNPAID"
            Me.txtReceived.Text = "0.00"
        Case "Paypal", "Cash", "Paypal eCheque", "Postal Order", "Cheque", "Other",
"eBay MP"
            Me.txtReceived.Text = lblOrderValueTotal.Text

    End Select

End Sub

Private Sub txtEbayUserID_TextChanged(sender As System.Object, e As System.EventArgs)
Handles txtEbayUserID.TextChanged
    On Error GoTo Err_Fail

    'Dont Run Code If eBay Name Is Blank
    If Trim(txtEbayUserID.Text) = "" Then Exit Sub

    'Look For Matching eBay User ID
    Dim intOrders As Integer
    intOrders = ReturnRecordCount("Select Ordernum from tblDeliveryAdds where
eBayName = '" & Trim(txtEbayUserID.Text) & "'")

    If intOrders > 0 Then
        Me.btnIMPORT.Enabled = True
        Me.btnIMPORT.BackColor = Color.LightGreen
        Me.btnIMPORT.Text = "IMPORT (" & Trim(Str(intOrders)) & ")"
    Else
        Me.btnIMPORT.Enabled = False
        Me.btnIMPORT.BackColor = Color.LightGray
        Me.btnIMPORT.Text = "IMPORT"

    End If

    Exit Sub
Err_Fail:
    MsgBox(ErrorToString)
End Sub

Private Sub btnSingleLabel_Click(sender As System.Object, e As System.EventArgs)
Handles btnSingleLabel.Click

    ' NOTE - New Address Label Table 'tblAddressLabels'

```

```

Dim sql As String
Dim con As New OleDb.OleDbConnection
Dim dbprovider As String
'Dim dbsource As String
dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
con.ConnectionString = dbprovider & dbsource
con.Open()

'Delete Existing Table Data
Dim upd As New OleDb.OleDbCommand
sql = "Delete * From tblAddressLabels"
upd = New OleDb.OleDbCommand(sql, con)
upd.ExecuteNonQuery()

'Insert Single Address Label Into Data Table
sql = "INSERT INTO tblAddressLabels ( AddressLabel, OrderNum, PostType ) SELECT
(tblDeliveryAdds.title & ' ' & tblDeliveryAdds.FirstName & ' ' & tblDeliveryAdds.Surname
& Chr(13) & Chr(10) & tblDeliveryAdds.Addr1 & Chr(13) & Chr(10) & tblDeliveryAdds.Addr2 &
', ' & tblDeliveryAdds.Addr3 & Chr(13) & Chr(10) & tblDeliveryAdds.Town & Chr(13) &
Chr(10) & tblDeliveryAdds.PCode) AS Expr1, '*' & tblDeliveryAdds.OrderNum & '*',
tblDeliveryAdds.Post_Method FROM tblDeliveryAdds WHERE tblDeliveryAdds.OrderNum = '" &
Me.lblOrderNumber.Text & "'"

Try
    upd = New OleDb.OleDbCommand(sql, con)
    upd.ExecuteNonQuery()
Catch ex As Exception
    MsgBox(ex.ToString)
End Try
con.Close()
upd.Dispose()

'Open Single (External) Label Report

'Process.Start("G:\Farout\Documents\Linked Documents\Dbase Single Label -
NET.lbx")
Process.Start(strAppPath & "Linked Documents\DbaseSingleLabel_NET.lbx")

End Sub

Private Function Build_MultiUseMenu() As Boolean
    'On Error GoTo Err_Fail    'Creates Lambda ???

    Dim con As New OleDb.OleDbConnection
    Dim dbprovider As String
    'Dim dbsource As String
    dbprovider = "PROVIDER=Microsoft.jet.OLEDB.4.0;"
    'dbsource = "Data Source = G:\Farout\dbFarout_Soft.mdb"
    con.ConnectionString = dbprovider & dbsource
    con.Open()
    Dim ds As New DataSet
    Dim da As OleDb.OleDbDataAdapter
    Dim sql As String

```

```

    sql = "SELECT TOP 5 tblDeliveryAdds.Pcode, Count(tblDeliveryAdds.Pcode) AS
[Count] From tblDeliveryAdds Where Len(tblDeliveryAdds.Pcode) > 0 GROUP BY
tblDeliveryAdds.Pcode ORDER BY Count(tblDeliveryAdds.Pcode) DESC"

    da = New OleDb.OleDbDataAdapter(sql, con)
    da.Fill(ds, "MyData")

    Dim MaxRecCount As Integer
    Dim inc As Integer
    inc = -1
    MaxRecCount = ds.Tables("MyData").Rows.Count

    Dim menu As New MenuStrip()

    For inc = 0 To MaxRecCount - 1

        'Dont Add Empty Options !!!

        Dim item As New ToolStripMenuItem(ds.Tables("MyData").Rows(inc).Item(0),
Nothing, AddressOf mnuMultiUse_Click)

        item.Tag = ds.Tables("MyData").Rows(inc).Item(0)

        mnuMultiUse.Items.Add(item)

    Next inc
    con.Close()

    Build_MultiUseMenu = True
    Exit Function
Err_Fail:
    MsgBox(ErrorToString)
    con.Close()
    Build_MultiUseMenu = False
    End Function

    Private Sub mnuMultiUse_Click(sender As Object, e As System.EventArgs) Handles
mnuMultiUse.Click

        mnuMultiUse.Hide() 'Sometimes the menu items can remain open. May not be
necessary for you.
        Dim item As ToolStripMenuItem = TryCast(sender, ToolStripMenuItem)
        If item IsNot Nothing Then
            Me.txtPCode.Text = item.Tag
        End If

    End Sub

    Private Function BuildPostDateMenu() As Boolean
        On Error GoTo Err_Fail

        Dim menu As New MenuStrip()

        'Today
        Dim item As New ToolStripMenuItem("Today", Nothing, AddressOf mnuPostDate_Click)

```

```

        item.Tag = "Today"
        mnuPostDate.Items.Add(item)

        'Tomorrow
        Dim item1 As New ToolStripMenuItem("Tomorrow", Nothing, AddressOf
mnuPostDate_Click)
        item1.Tag = "Tomorrow"
        mnuPostDate.Items.Add(item1)

        'Next Working Day
        Dim item2 As New ToolStripMenuItem("Next Working Day", Nothing, AddressOf
mnuPostDate_Click)
        item2.Tag = "Next Working Day"
        mnuPostDate.Items.Add(item2)

        'Yesterday
        Dim item3 As New ToolStripMenuItem("Yesterday", Nothing, AddressOf
mnuPostDate_Click)
        item3.Tag = "Yesterday"
        mnuPostDate.Items.Add(item3)

        'Same As Order Date
        Dim item4 As New ToolStripMenuItem("Same As Order Date", Nothing, AddressOf
mnuPostDate_Click)
        item4.Tag = "Same As Order Date"
        mnuPostDate.Items.Add(item4)

        BuildPostDateMenu = True
        Exit Function
Err_Fail:

        MsgBox(ErrorToString)
        BuildPostDateMenu = False

    End Function

    Private Sub mnuPostDate_Click(sender As Object, e As System.EventArgs) Handles
mnuPostDate.Click

        mnuPostDate.Hide() 'Sometimes the menu items can remain open. May not be
necessary for you.

        Dim item As ToolStripMenuItem = TryCast(sender, ToolStripMenuItem)

        If item IsNot Nothing Then

            Select Case item.Tag

                Case "Today"
                    Me.txtPostDate.Text = Date.Today
                Case "Tomorrow"
                    Me.txtPostDate.Text = DateAdd(DateInterval.Day, 1, Date.Today)
                Case "Next Working Day"
                    Me.txtPostDate.Text = NextWorkingDay()
                Case "Yesterday"
                    Me.txtPostDate.Text = DateAdd(DateInterval.Day, -1, Date.Today)
            End Select
        End If
    End Sub

```

```
        Case "Same As Order Date"  
            Me.txtPostDate.Text = lblOrderDate.Text  
        Case Else  
            Me.txtPostDate.Text = Me.txtPostDate.Text  
    End Select
```

```
        'Me.txtPostDate.Text = item.Tag
```

```
    End If
```

```
End Sub
```

```
Private Sub btnClearBarcode_Click(sender As System.Object, e As System.EventArgs) Handles btnClearBarcode.Click
```

```
    Me.txtOrderBarCode.Text = ""
```

```
    Me.txtOrderBarCode.Focus()
```

```
End Sub
```

```
Private Sub txtOrderBarCode_TextChanged(sender As System.Object, e As System.EventArgs) Handles txtOrderBarCode.TextChanged
```

```
    Dim strBarcode As String  
    strBarcode = Trim(Me.txtOrderBarCode.Text)
```

```
    If IsValidOrderNumber(strBarcode) = True Then
```

```
        'MsgBox("open order : " & strBarcode)
```

```
        Me.MdiParent = Me.MdiParent
```

```
        Me.Show()
```

```
        Me.LoadExistingOrder(strBarcode)
```

```
    End If
```

```
End Sub
```

```
Private Sub lblEMail_DoubleClick(sender As Object, e As System.EventArgs) Handles lblEMail.DoubleClick
```

```
    Clipboard.SetText(Me.txtTo.Text)
```

```
    Me.lblEMail.ForeColor = Color.Green
```

```
End Sub
```

```

Private Sub tabCurrentOrder_DrawItem(sender As Object, e As
System.Windows.Forms.DrawItemEventArgs) Handles tabCurrentOrder.DrawItem
    ' Identify which TabPage is currently selected
    Dim SelectedTab As TabPage = tabCurrentOrder.TabPages(e.Index)

    ' Get the area of the header of this TabPage
    Dim HeaderRect As Rectangle = tabCurrentOrder.GetTabRect(e.Index)

    ' Create two Brushes to paint the Text
    Dim BlackTextBrush As New SolidBrush(Color.Black)
    Dim RedTextBrush As New SolidBrush(Color.DarkOrange)

    ' Set the Alignment of the Text
    Dim sf As New StringFormat()
    sf.Alignment = StringAlignment.Center
    sf.LineAlignment = StringAlignment.Center

    ' Paint the Text using the appropriate Bold and Color setting
    If Convert.ToBoolean(e.State And DrawItemState.Selected) Then

        Dim BoldFont As New Font(tabCurrentOrder.Font.Name,
tabCurrentOrder.Font.Size, FontStyle.Regular)

        e.Graphics.DrawString(SelectedTab.Text, BoldFont, RedTextBrush, HeaderRect,
sf)

    Else
        e.Graphics.DrawString(SelectedTab.Text, e.Font, BlackTextBrush, HeaderRect,
sf)
    End If

    ' Job done - dispose of the Brushes
    BlackTextBrush.Dispose()
    RedTextBrush.Dispose()
End Sub

Private Sub btnNudge_Click(sender As System.Object, e As System.EventArgs) Handles
btnNudge.Click
    On Error GoTo Err_Fail

    'Detect Number Only In Addr1

    If IsNumeric((Trim(txtAddr1.Text))) Then
        txtAddr1.Text = Trim(txtAddr1.Text) & " " & txtAddr2.Text
        txtAddr2.Text = txtAddr3.Text
        txtAddr3.Text = txtTown.Text
        txtTown.Text = txtPCode.Text
        txtPCode.Text = ""
    End If

    'Nudge Up Address Details Into Empty Lines
    If Len(Trim(txtAddr1.Text)) = 0 Then
        txtAddr1.Text = txtAddr2.Text
        txtAddr2.Text = txtAddr3.Text
        txtAddr3.Text = txtTown.Text
        txtTown.Text = txtPCode.Text
        txtPCode.Text = ""
    End If

```



```

    If Len(Trim(txtAddr2.Text)) = 0 Then
        txtAddr2.Text = txtAddr3.Text
        txtAddr3.Text = txtTown.Text
        txtTown.Text = txtPCode.Text
        txtPCode.Text = ""
    End If
    If Len(Trim(txtAddr3.Text)) = 0 Then
        txtAddr3.Text = txtTown.Text
        txtTown.Text = txtPCode.Text
        txtPCode.Text = ""
    End If
    If Len(Trim(txtTown.Text)) = 0 Then
        txtTown.Text = txtPCode.Text
        txtPCode.Text = ""
    End If

    'Is Postcode In Final Field? If So Nudge It Up...
    If Len(Trim(txtPCode.Text)) > 0 Then
        If IsPostCode(Me.txtPCode.Text) = True Then

            Me.txtAddr3.Text = Me.txtAddr3.Text & " " & Me.txtTown.Text

            Me.txtTown.Text = Me.txtPCode.Text

            Me.txtPCode.Text = ""

        End If
    End If

    Exit Sub
Err_Fail:
    MsgBox(ErrorToString)

End Sub

Private Sub btnStock_Click(sender As System.Object, e As System.EventArgs) Handles
btnStock.Click
    On Error GoTo Err_Fail
    'Verify Stock Code
    Dim strStockCode As String = Trim(Me.lblStockCode.Text)
    If Mid(strStockCode, 1, 1) = "S" And Len(strStockCode) = 9 Then
        'Open Stock Form
        frmStock.MdiParent = mdiShell
        frmStock.Show()
        'Open Specified Stock Details
        Call frmStock.FillStockForm(Me.lblStockCode.Text)
    End If
    Exit Sub
Err_Fail:
    MsgBox(ErrorToString)
End Sub

```

End Class