

```
'Allows the user to make inter-account transfers by UPDATING the paying and collecting
accounts'
'records and INSERTING transactions for both the paying and collecting accounts in
'the Transaction table of the FAMSDB
'Written by Bill Blackley
'First written 25/8/2005
'Last written 15/9/2005
```

```
Public Class frmInterAccountTransfer
    Inherits System.Windows.Forms.Form
```

```
Windows Form Designer generated code
```

```
'Allows the user to transfer funds from the account to another account and
'INSERTs a record in the Transaction Tble of the FAMSDB database
Private Sub cmdTransfer_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles cmdTransfer.Click
```

```
Try
```

```
    'First Check that receiving (collecting) account exists
```

```
    'Open database connection
    FAMSDBConnection.Open()
```

```
    'Return Dataset To It's Original state
    AccountDataSet.Reset()
```

```
    'Create SELECT query to check the collecting account exists
    AccountDataAdapter.SelectCommand.CommandText = _
    "SELECT * FROM account WHERE sortCode = '" & _
    txtCollectingSortCode.Text & "' AND accountNo = '" & _
    txtCollectingAccountNo.Text & "'"
```

```
    'Fill DataSet
    AccountDataAdapter.Fill(AccountDataSet)
```

```
    'Close Database Connection
    FAMSDBConnection.Close()
```

```
    'Put results of query into datatable with which to work
    Dim datatable As DataTable = AccountDataSet.Tables(0)
```

```
    'Check that record exists
    If datatable.Rows.Count <> 0 Then
```

```
        'Declare variables to hold the amounts being altered by the transaction
        Dim balance, newBalance, availableBalance, newAvailableBalance _
        As String
```

```
        'Open Database Connection
        FAMSDBConnection.Open()
```

```
        'Return DataSet to it's original state
        AccountDataSet.Reset()
```

```
        'Create SELECT query to get balance and availableBalance values
        'from Paying account
        AccountDataAdapter.SelectCommand.CommandText = _
        "SELECT balance, availableBalance FROM Account WHERE sortcode = '" & _
        txtSortCode.Text & "' AND accountNo = '" & txtAccountNo.Text & "'"
```

```
        'Fill DataSet
        AccountDataAdapter.Fill(AccountDataSet)
```

```
        'Close Database Connection
        FAMSDBConnection.Close()
```

```
        'Put results of query into datatable with which to work
        Dim datatable1 As DataTable = AccountDataSet.Tables(0)
```

```
        'Put results of query into balance and availableBalance variables
        balance = Convert.ToString(datatable1.Rows(0)(0))
        availableBalance = Convert.ToString(datatable1.Rows(0)(1))
```

```

'Check that there are sufficient funds available
If Convert.ToDecimal(txtAmount.Text) <= Convert.ToDecimal(availableBalance)
Then
    'First UPDATE Record of Paying Account

    'Now calculate new balances after money has been transfered
    newBalance = Convert.ToString(Convert.ToDecimal(balance) - Convert.
ToDecimal(txtAmount.Text))
    newAvailableBalance = Convert.ToString(Convert.ToDecimal
(availableBalance) - Convert.ToDecimal(txtAmount.Text))

    'Open Database Connection
    FAMSDBConnection.Open()

    'Create UPDATE Command to update paying account balances
    AccountDataAdapter.UpdateCommand.CommandText = _
    "UPDATE Account SET balance = '" & newBalance & "', " & _
    "availableBalance = '" & newAvailableBalance & "' " & _
    "WHERE sortCode = '" & txtSortCode.Text & "' AND " & _
    "accountNo = '" & txtAccountNo.Text & "'"

    'Execute UPDATE Command
    AccountDataAdapter.UpdateCommand.ExecuteNonQuery()

    'Close Database Connection
    FAMSDBConnection.Close()

    'Now Insert Transaction for Paying account
    Transaction.InsertTransaction("PayTransfer", txtAmount.Text, newBalance
txtSortCode.Text, txtAccountNo.Text)

    'Now UPDATE Record of receiving account

    'Open Database Connection
    FAMSDBConnection.Open()

    'Return DataSet to it's original state
    AccountDataSet.Reset()

    'Create SELECT query to get receiving accounts balances
    AccountDataAdapter.SelectCommand.CommandText = _
    "SELECT balance, availableBalance FROM account WHERE sortCode = '" & _
    txtCollectingSortCode.Text & "' AND accountNo = '" & _
    txtCollectingAccountNo.Text & "'"

    'Fill DataSet
    AccountDataAdapter.Fill(AccountDataSet)

    'Close Database Connection
    FAMSDBConnection.Close()

    'Put results of query into datatable with which to work
    Dim datatable2 As DataTable = AccountDataSet.Tables(0)

    'Put results of query into balance and availableBalance variables
    balance = Convert.ToString(datatable2.Rows(0)(0))
    availableBalance = Convert.ToString(datatable2.Rows(0)(1))

    'Now calculate new balances
    newBalance = Convert.ToString(Convert.ToDecimal(balance) + Convert.
ToDecimal(txtAmount.Text))
    newAvailableBalance = Convert.ToString(Convert.ToDecimal
(availableBalance) + Convert.ToDecimal(txtAmount.Text))

    'Open Database Connection
    FAMSDBConnection.Open()

    'Create UPDATE Command to update collecting account balances
    AccountDataAdapter.UpdateCommand.CommandText = _
    "UPDATE Account SET balance = '" & newBalance & "', " & _
    "availableBalance = '" & newAvailableBalance & "' " & _
    "WHERE sortCode = '" & txtCollectingSortCode.Text & "' AND " & _

```

```
        "accountNo = '" & txtCollectingAccountNo.Text & "'"

        'Execute UPDATE Command
        AccountDataAdapter.UpdateCommand.ExecuteNonQuery()

        'Close Database Connection
        FAMSDBConnection.Close()

        'Now Insert Transaction for Collecting account
        Transaction.InsertTransaction("ReceiveTransfer", txtAmount.Text,
newBalance, _
        txtCollectingSortCode.Text, txtCollectingAccountNo.Text)

        'Display Confimrtaion of transfer message
        MessageBox.Show("£" & txtAmount.Text & " has been transfered",
"Transfer Complete", MessageBoxButtons.OK, MessageBoxIcon.Information)

        'Close the form
        Me.Close()
    Else
        'Display warning message if there are insufficient funds for the
transfer
        MessageBox.Show("Insufficient Funds Available", "Insufficient Funds",
        MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Exit Sub
    End If

    Else
        'Display warning message if the collecting account does not exist
        MessageBox.Show("Collecting Account Not Found", "Account Not Found",
        MessageBoxButtons.OK, MessageBoxIcon.Warning)
    End If

    Catch ex As Exception
        MessageBox.Show(ex.ToString())
    End Try

End Sub 'cmdTransfer_Click

End Class 'frmInterAccountTransfer
```