

Chapter 15 Records and Random-Access Files **(Main Page)**

- 15.1 Visual Basic's view of a random-access file.
- 15.2 Access types available for random-access files.
- 15.3 A possible storage alignment for a variable type **Example** showing an undefined area in memory.
- 15.4 Writing empty records to a file.
- 15.5 Writing data randomly to a random-access file.
- 15.6 Reading data sequentially from a random-access file.
- 15.7 Reading randomly from a random-access file.
- 15.8 A transaction processing program.

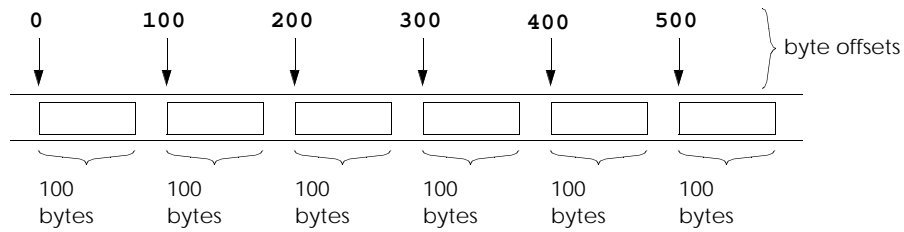


Fig. 15.1 Visual Basic's view of a random-access file.

Access type	Description
Read	Open the file in read-only mode.
Write	Open the file in write-only mode.
Read Write	Open the file for reading and writing.

Fig. 15.2 Access types available for random-access files.

Byte	0	1	2	3
	01100001		00000000	01100001

Fig. 15.3 A possible storage alignment for a variable of type **Example** showing an undefined area in memory.

```

1  ' Fig. 15.4
2  ' Creating a blank random-access file
3  Option Explicit
4
5  Private Type ClientRecord
6      accountNumber As Integer
7      lastName As String * 15
8      firstName As String * 15
9      balance As Currency
10 End Type
11
12 Sub cmdInitialize_Click()
13     Dim recordLength As Long, x As Integer
14     Dim udtBlankClient As ClientRecord ' user-defined type
15     Dim filename As String
16
17     ' Determine number of bytes in a ClientRecord object
18     recordLength = LenB(udtBlankClient)
19
20     dlgOpen.ShowOpen
21     filename = dlgOpen.filename
22
23     If dlgOpen.FileTitle <> "" Then
24         ' Open clients.rnd for writing

```

```

25      Open filename For Random Access Write As #1 _
26          Len = recordLength
27
28      For x = 1 To 100
29          Put #1, x, udtBlankClient ' Write empty records
30      Next
31
32      Close #1 ' Close file
33
34      cmdInitialize.Enabled = False ' Disable button
35      MsgBox ("File initialized. Click Exit to terminate.")
36  Else
37      MsgBox ("You must specify a file name")
38  End If
39 End Sub
40
41 Sub cmdExit_Click()
42     End
43 End Sub

```

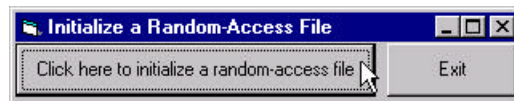


Fig. 15.4 Writing empty records to a file (part 1 of 2).

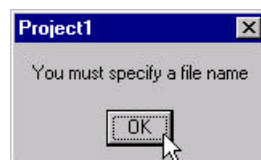
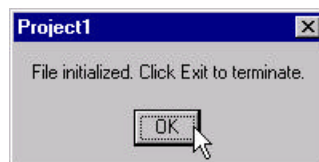
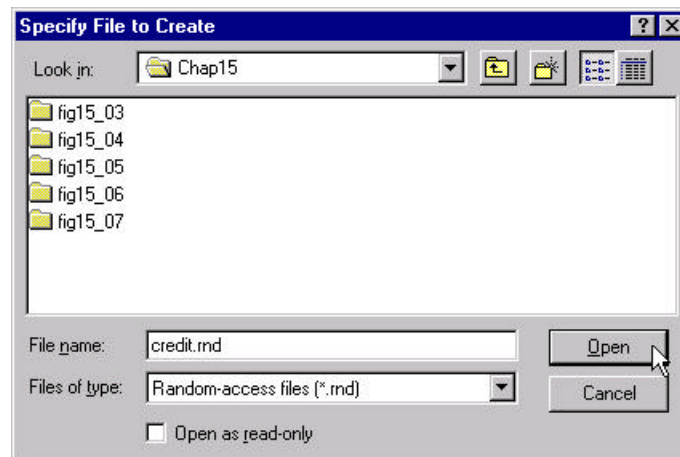


Fig. 15.4 Writing empty records to a file (part 2 of 2).

```

1  ' Fig. 15.5
2  ' Writing data to a random-access file
3  Option Explicit
4
5  Private Type ClientRecord
6      accountNumber As Integer
7      lastName As String * 15
8      firstName As String * 15
9      balance As Currency
10 End Type
11
12 Dim mUdtClient As ClientRecord    ' user-defined type
13
14 Private Sub Form_Load()
15     cmdEnter.Enabled = False
16     cmdDone.Enabled = False
17 End Sub
18
19 Sub cmdOpenFile_Click()
20     Dim recordLength As Long
21     Dim filename As String
22
23     ' Determine number of bytes in a ClientRecord object
24     recordLength = LenB(mUdtClient)
25
26     dlgOpen.ShowOpen
27     filename = dlgOpen.filename
28
29     If dlgOpen.FileTitle <> "" Then
30         ' Open file for writing
31         Open filename For Random Access Write As #1 _
32             Len = recordLength
33
34         cmdOpenFile.Enabled = False    ' Disable button
35         cmdEnter.Enabled = True

```

Fig. 15.5 Writing data randomly to a random-access file (part 1 of 3).

```

36     cmdDone.Enabled = True
37 Else
38     MsgBox ("You must specify a file name")
39 End If
40 End Sub
41
42 Private Sub cmdEnter_Click()
43     mUdtClient.accountNumber = Val(txtAccount.Text)
44     mUdtClient.firstName = txtFirstName.Text
45     mUdtClient.lastName = txtLastName.Text
46     mUdtClient.balance = Val(txtBalance.Text)
47
48     ' Write record to file
49     Put #1, mUdtClient.accountNumber, mUdtClient
50
51     Call ClearFields
52 End Sub
53
54 Sub cmdDone_Click()
55     Close #1
56     cmdOpenFile.Enabled = True
57     cmdEnter.Enabled = False
58     cmdDone.Enabled = False
59 End Sub

```

```

60
61 Private Sub Form_Terminate()
62     Close #1
63 End Sub
64
65 Private Sub ClearFields()
66     txtAccount.Text = ""
67     txtFirstName.Text = ""
68     txtLastName.Text = ""
69     txtBalance.Text = ""
70 End Sub

```



Fig. 15.5 Writing data randomly to a random-access file (part 2 of 3).

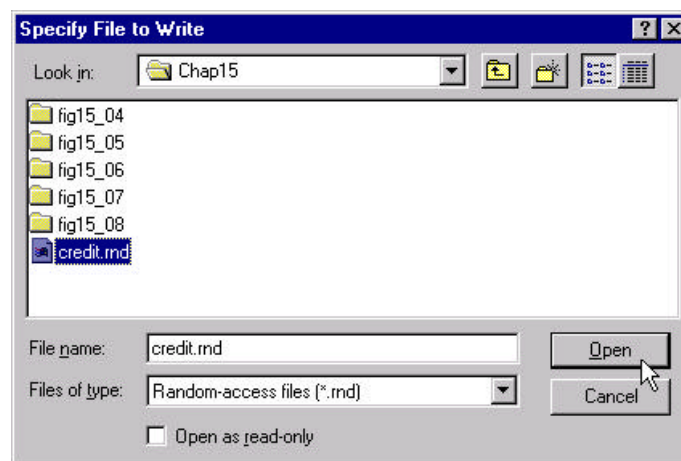


Fig. 15.5 Writing data randomly to a random-access file (part 3 of 3).

```

1  ' Fig. 15.6
2  ' Reading data sequentially from a random-access file
3  Option Explicit
4
5  Private Type ClientRecord
6      accountNumber As Integer
7      lastName As String * 15
8      firstName As String * 15
9      balance As Currency
10 End Type
11
12 Dim mUdtClient As ClientRecord    ' user-defined type
13
14 Private Sub Form_Load()
15     cmdNext.Enabled = False
16     cmdDone.Enabled = False
17 End Sub
18
19 Sub cmdOpenFile_Click()
20     Dim recordLength As Long
21     Dim filename As String
22
23     ' Determine number of bytes in a ClientRecord object
24     recordLength = LenB(mUdtClient)
25
26     dlgOpen.ShowOpen
27     filename = dlgOpen.filename
28
29     If dlgOpen.FileTitle <> "" Then
30         ' Open file for writing
31         Open filename For Random Access Read As #1 _
32             Len = recordLength
33         cmdOpenFile.Enabled = False    ' Disable button
34         cmdNext.Enabled = True

```

Fig. 15.6 Reading data sequentially from a random-access file (part 1 of 3).

```

35     cmdDone.Enabled = True
36 Else
37     MsgBox ("You must specify a file name")
38 End If
39 End Sub
40
41 Private Sub cmdNext_Click()
42     ' Read record from file
43     Do
44         Get #1, , mUdtClient
45     Loop Until EOF(1) Or mUdtClient.accountNumber <> 0
46
47     If EOF(1) Then
48         cmdNext.Enabled = False
49         Exit Sub
50     End If
51
52     If mUdtClient.accountNumber <> 0 Then
53         txtAccount.Text = Str$(mUdtClient.accountNumber)
54         txtFirstName.Text = mUdtClient.firstName
55         txtLastName.Text = mUdtClient.lastName
56         txtBalance.Text = Str$(mUdtClient.balance)
57     End If

```

```

58 End Sub
59
60 Sub cmdDone_Click()
61     Close #1
62     cmdOpenFile.Enabled = True
63     cmdNext.Enabled = False
64     cmdDone.Enabled = False
65     txtAccount.Text = ""
66     txtFirstName.Text = ""
67     txtLastName.Text = ""
68     txtBalance.Text = ""
69 End Sub
70
71 Private Sub Form_Terminate()
72     Close #1
73 End Sub

```



Fig. 15.6 Reading data sequentially from a random-access file (part 2 of 3).

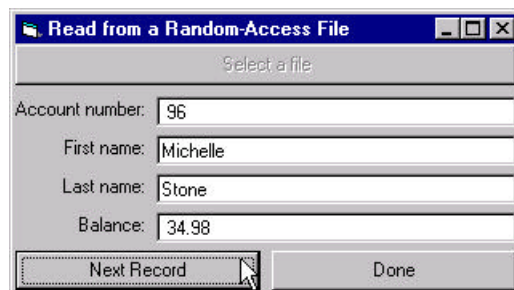
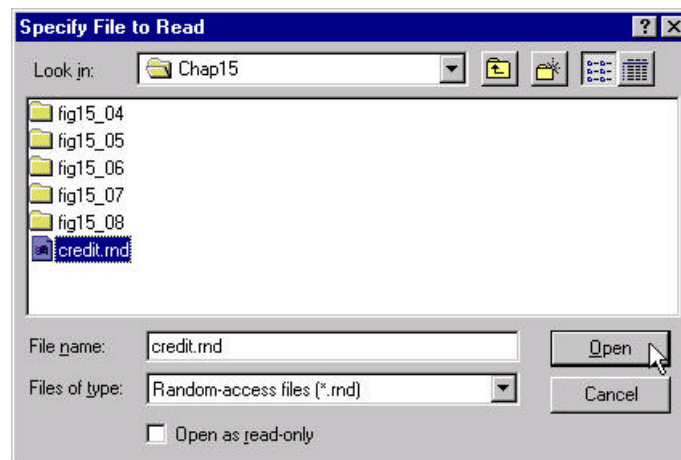


Fig. 15.6 Reading data sequentially from a random-access file (part 3 of 3).

```

1  ' Fig. 15.7
2  ' Reading data randomly from a random-access file
3  Option Explicit
4
5  Private Type ClientRecord
6      accountNumber As Integer
7      lastName As String * 15
8      firstName As String * 15
9      balance As Currency
10 End Type
11
12 Dim mUdtClient As ClientRecord    ' user-defined type
13
14 Private Sub Form_Load()
15     cmdGet.Enabled = False
16     cmdDone.Enabled = False
17 End Sub

```

Fig. 15.7 Reading randomly from a random-access file (part 1 of 4).

```

18
19 Sub cmdOpenFile_Click()
20     Dim recordLength As Long
21     Dim filename As String
22
23     ' Determine number of bytes in a ClientRecord object
24     recordLength = LenB(mUdtClient)
25
26     dlgOpen.ShowOpen
27     filename = dlgOpen.filename
28
29     If dlgOpen.FileTitle <> "" Then
30         ' Open file for writing
31         Open filename For Random Access Read As #1 _
32             Len = recordLength
33         cmdOpenFile.Enabled = False    ' Disable button
34         cmdGet.Enabled = True
35         cmdDone.Enabled = True
36     Else
37         MsgBox ("You must specify a file name")
38     End If
39 End Sub
40
41 Private Sub cmdGet_Click()
42     On Error Resume Next
43     ' Read record from file
44     If txtAccount.Text <> "" Then
45         Get #1, Val(txtAccount.Text), mUdtClient
46
47         If mUdtClient.accountNumber <> 0 Then
48             txtAccount.Text = Str$(mUdtClient.accountNumber)
49             txtFirstName.Text = mUdtClient.firstName
50             txtLastName.Text = mUdtClient.lastName
51             txtBalance.Text = Str$(mUdtClient.balance)
52         ElseIf mUdtClient.accountNumber = 0 Then
53             txtFirstName.Text = "Record not found"
54             txtLastName.Text = ""
55             txtBalance.Text = ""
56         End If
57     Else
58         MsgBox ("You must specify an Account Number")

```



```

59     End If
60 End Sub
61
62 Sub cmdDone_Click()
63     Close #1
64     cmdOpenFile.Enabled = True
65     cmdGet.Enabled = False
66     cmdDone.Enabled = False
67     txtAccount.Text = ""
68     txtFirstName.Text = ""
69     txtLastName.Text = ""

```

Fig. 15.7 Reading randomly from a random-access file (part 2 of 4).

```

70     txtBalance.Text = ""
71 End Sub
72
73 Private Sub Form_Terminate()
74     Close #1
75 End Sub

```

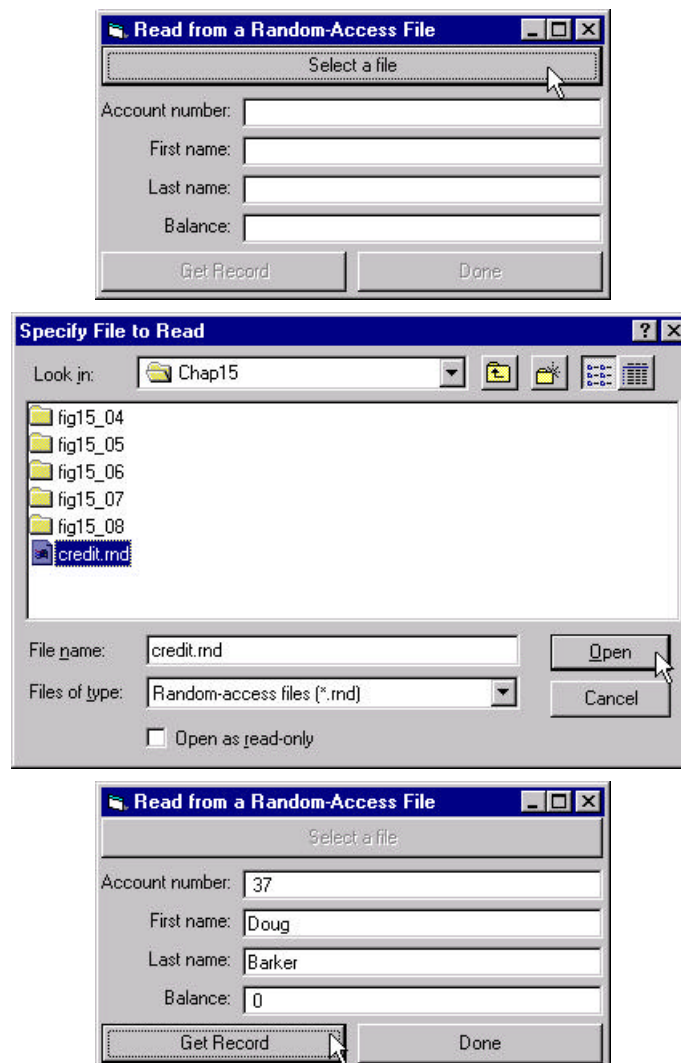


Fig. 15.7 Reading randomly from a random-access file (part 3 of 4).



Fig. 15.7 Reading randomly from a random-access file (part 4 of 4).

```

1  ' Fig. 15.8
2  ' Transaction processing program with random-access files
3  Option Explicit
4
5  Private Type ClientRecord
6      accountNumber As Integer
7      lastName As String * 15
8      firstName As String * 15
9      balance As Currency
10 End Type
11
12 Dim mUdtClient As ClientRecord    ' user-defined type
13
14 Private Sub Form_Load()
15     tabOperations.Enabled = False
16 End Sub

```

Fig. 15.8 A transaction processing program (part 1 of 4).

```

17
18 Sub cmdOpenFile_Click()
19     Dim recordLength As Long
20     Dim filename As String
21
22     ' Determine number of bytes in a ClientRecord object
23     recordLength = LenB(mUdtClient)
24
25     dlgOpen.ShowOpen
26     filename = dlgOpen.filename
27
28     If dlgOpen.FileTitle <> "" Then
29         ' Open file for writing
30         Open filename For Random Access Read Write As #1 _
31             Len = recordLength
32         cmdOpenFile.Enabled = False    ' Disable button
33         cmdCloseFile.Enabled = True
34         tabOperations.Enabled = True
35     Else
36         MsgBox ("You must specify a file name")
37     End If
38 End Sub
39
40 ' Create a text file representation of the random-access file
41 Private Sub cmdTextFile_Click()
42     Dim filename As String, balanceString As String
43

```

```

44 On Error Resume Next
45 dlgTextFile.ShowOpen
46 filename = dlgTextFile.filename
47
48 If dlgTextFile.FileTitle <> "" Then
49     ' Open file for writing
50     Open filename For Output Access Write As #2
51     Print #2, "Account";
52     Print #2, Tab(10); "First Name";
53     Print #2, Tab(28); "Last Name";
54     Print #2, Tab(46); Format("Balance", "#####")
55
56     Seek #1, 1           ' reposition to start of file
57     Get #1, , mUdtClient ' read first record
58
59     While Not EOF(1)
60         If mUdtClient.accountNumber <> 0 Then
61             Print #2, mUdtClient.accountNumber;
62             Print #2, Tab(10); mUdtClient.firstName;
63             Print #2, Tab(28); mUdtClient.lastName;
64             balanceString = _
65                 Format(mUdtClient.balance, "0.00")
66             Print #2, Tab(46);
67             Print #2, Format(balanceString, "#####")
68
69         End If

```

Fig. 15.8 A transaction processing program (part 2 of 4).

```

70
71     Get #1, , mUdtClient 'read next record
72     Wend
73
74     Close #2
75 Else
76     MsgBox ("You must specify a file name")
77 End If
78 End Sub
79
80 ' Add a new record to the file
81 Private Sub cmdAddNew_Click()
82     If txtNewAccount.Text <> "" Then
83         Get #1, Val(txtNewAccount), mUdtClient 'read record
84
85         If mUdtClient.accountNumber = 0 Then
86             mUdtClient.accountNumber = Val(txtNewAccount)
87             mUdtClient.firstName = txtNewFirstName.Text
88             mUdtClient.lastName = txtNewLastName.Text
89             mUdtClient.balance = txtNewBalance.Text
90             Put #1, mUdtClient.accountNumber, mUdtClient
91             MsgBox ("Account " & mUdtClient.accountNumber & _
92                 " has been added to the file")
93         Else
94             MsgBox ("Account already exists")
95         End If
96     Else
97         MsgBox ("You must enter an account number")
98     End If
99 End Sub
100
101 ' Update an existing record
102 Private Sub cmdUpdate_Click()

```

```

103 Dim account As Integer, transactionAmount As Double
104 On Error Resume Next
105
106 account = Val(InputBox("Enter account number"))
107 Get #1, account, mUdtClient 'read record
108
109 If mUdtClient.accountNumber <> 0 Then
110     txtUpdateAccount.Text = Str$(mUdtClient.accountNumber)
111     txtUpdateFirstName.Text = mUdtClient.firstName
112     txtUpdateLastName.Text = mUdtClient.lastName
113     txtUpdateBalance.Text = Str$(mUdtClient.balance)
114     transactionAmount = Val(InputBox( _
115         "Enter transaction amount. Positive for charge. " & _
116         "Negative for payment."))
117     mUdtClient.balance = _
118         mUdtClient.balance + transactionAmount
119     txtUpdateBalance.Text = Str$(mUdtClient.balance)
120     Put #1, mUdtClient.accountNumber, mUdtClient

```

Fig. 15.8 A transaction processing program (part 3 of 4).

```

121 Else
122     MsgBox ("Record " & account & " does not exist")
123 End If
124
125 End Sub
126
127 ' Delete the specified record
128 Private Sub cmdDelete_Click()
129     Dim blankClient As ClientRecord
130     On Error Resume Next
131
132     Get #1, Val(txtDelete.Text), mUdtClient 'read record
133
134     If mUdtClient.accountNumber <> 0 Then
135         Put #1, mUdtClient.accountNumber, blankClient
136         MsgBox ("Account # " & mUdtClient.accountNumber & _
137             " has been deleted")
138     Else
139         MsgBox ("Record does not exist")
140     End If
141 End Sub
142
143 Sub cmdCloseFile_Click()
144     Close #1
145     cmdOpenFile.Enabled = True
146     cmdCloseFile.Enabled = False
147 End Sub
148
149 Private Sub Form_Terminate()
150     Close
151 End Sub
152
153 Private Sub cmdExit_Click()
154     Close
155     End
156 End Sub

```

Fig. 15.8 A transaction processing program (part 4 of 4).

