

Minorbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

1 Minor Breaks
Please use speaker notes for additional information!

2 Minor Total Processing

3

4

5

6

Minor Total Processing

These notes are old and the flowchart was for a program written in COBOL, but it conveys the logic.

Input/Output

This is the input data - each record on the file contains a dept # and an amount.

Dept	Amount
12	500
12	100
12	50
15	200
15	100
15	300
15	60
17	20
17	240

Output report that is produced the input records and a total each time the department number changed.

DEPT #	AMOUNT
12	500
12	100
12	50
Total 12: 650	
15	200
15	100
15	300
15	60
Total 15: 660	
17	20
17	240
Total 17: 260	
Final Total: 1570	

Records from department 12.

Total for department 12.

Final total for all departments.

The input file is in order by dept #. When the dept # changes, we want to print out a total for that dept.

Slide 2 of 9

Comments

12:32 PM 2/20/2020

Minorbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

1 Minor Breaks
Please use speaker notes for additional information?

2 Minor Total Processing

3

4

5

6

Process second record:

When the second record is read, the DEPT is compared to DEPT-HOLD (12=12) so the AMOUNT is added to DEPT-TOTAL and FINAL-TOTAL, the line is written and the third input record is read.

The second input record: DEPT AMOUNT
12 100

DEPT #	AMOUNT
12	500
12	100

DEPT-HOLD: 12

DEPT-TOTAL (MINOR): ~~500~~ 600

FINAL-TOTAL: ~~500~~ 600

When the second record is processed, there is no break because the DEPT is equal to the DEPT-HOLD. Therefore the amounts are added to the accumulators and the line is written.

Slide 4 of 9

Type here to search

Comments

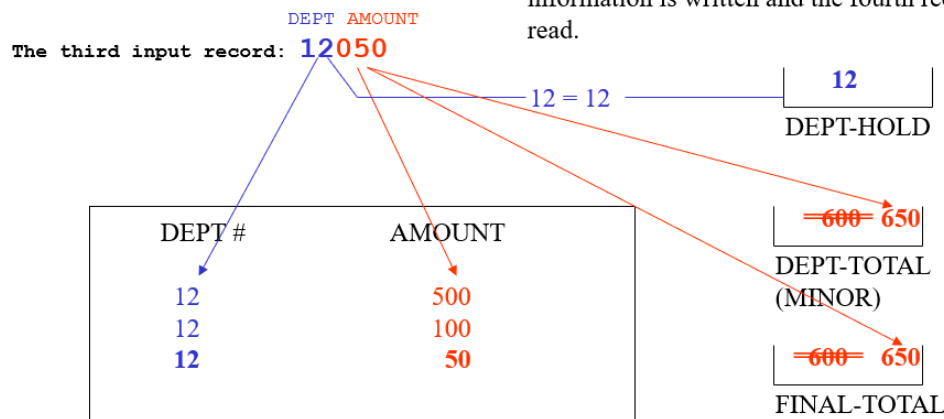
113%

12:34 PM
2/20/2020

- 1 Minor Breaks
Please use speaker notes for additional information!
- 2 Minor Total Processing
- 3 When the first input record is read, the DEPT is compared to DEPT-HOLD. If DEPT-HOLD is equal to DEPT, then the amount is added to DEPT-TOTAL (MINOR) and the amount is added to the accumulator and the detail line is written.
- 4 When the second record is read, the DEPT is compared to DEPT-HOLD. If DEPT-HOLD is equal to DEPT, then the amount is added to DEPT-TOTAL (MINOR) and the amount is added to the accumulator and the detail line is written.
- 5 When the third record is read, the DEPT is compared to DEPT-HOLD. If DEPT-HOLD is equal to DEPT, then the amount is added to DEPT-TOTAL (MINOR) and the amount is added to the accumulator and the detail line is written.
- 6 When the fourth record is read, the DEPT is compared to DEPT-HOLD. If DEPT-HOLD is not equal to DEPT, then the amount is added to DEPT-TOTAL (MINOR) and the amount is added to the accumulator and the detail line is written.

Process third record:

When the third record is read the DEPT is compared to the HOLD-DEPT (12 = 12). They are equal, so the amount is added to the DEPT-TOTAL and the FINAL-TOTAL. The information is written and the fourth record is read.



The third record is also from the same department so there is no break. The amount is added to the accumulators and the detail line is written.

Minorbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

Process fourth record:

When the fourth record is read, DEPT is compared to DEPT-HOLD (15 NOT = 12), therefore a DEPT break has occurred.

- DEPT-TOTAL moved to minor total line and minor total line is written
- RESET DEPT-HOLD to new DEPT & DEPT-TOTAL to 0

Process the record that caused the break by adding to DEPT-TOTAL and FINAL-TOTAL and writing the line. Then the fifth record is read.

The fourth input record: **15** 200

15 not = 12 so break

DEPT #	AMOUNT
12	500
12	100
12	50
Total 12:	650
15	200

DEPT-HOLD: ~~12~~ 15

DEPT-TOTAL (MINOR): ~~600 650 0~~ 200

FINAL-TOTAL: ~~600 650~~ 850

The fourth record causes a break because DEPT on the fourth record is not = DEPT-HOLD. The MINOR-TOTAL is processed which means:

1. The DEPT-TOTAL which is the minor total is moved to the

Minorbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

```

    graph TD
      B300DETAIL([B-300-DETAIL]) --> NeedHeader{Need header?}
      NeedHeader -- N --> SetDetail[Set up detail line]
      NeedHeader -- Y --> B400HDRROUT[B-400-HDR-ROUT]
      B400HDRROUT --> SetDetail
      SetDetail --> WriteLine[/WRITE LINE/]
      WriteLine --> AddTotals[Add to MINOR-TOTAL  
FINAL-TOTAL]
      AddTotals --> EndB300([END B-300])

      B310MINORROUT([B-310-MINOR-ROUT]) --> SetMinorTotal[Set up minor total line]
      SetMinorTotal --> WriteMinorTotal[/WRITE MINOR TOTAL LINE/]
      WriteMinorTotal --> ResetDept[RESET:  
DEPT to DEPT-HOLD]
      ResetDept --> ResetMinorTotal[RESET:  
MINOR-TOTAL = 0]
      ResetMinorTotal --> EndB310([END B-310])

      B320FINALROUT([B-320-FINAL ROUT]) --> SetFinalTotal[Set up final total line]
      SetFinalTotal --> WriteFinalTotal[/WRITE FINAL TOTAL LINE/]
      WriteFinalTotal --> EndB320([END B-320])

      B400HDRROUT([B-400-HDR-ROUT]) --> SetPageHeaders[Set up page and line headers]
      SetPageHeaders --> WritePageHeaders[/WRITE PAGE & LINE HEADERS/]
      WritePageHeaders --> AddPage[Add to page # and reset line count]
      AddPage --> EndB400([END B-400])

      C100WRAPUP([C-100-WRAPUP]) --> CloseFiles[CLOSE FILES]
      CloseFiles --> EndC100([END C-100])
  
```

MINOR TOTAL LOGIC FLOWCHART
Page 2

This is page 2 of the logic flowchart. Again, for illustration purposes, I am assuming that the MINOR break is caused by a change in DEPT.

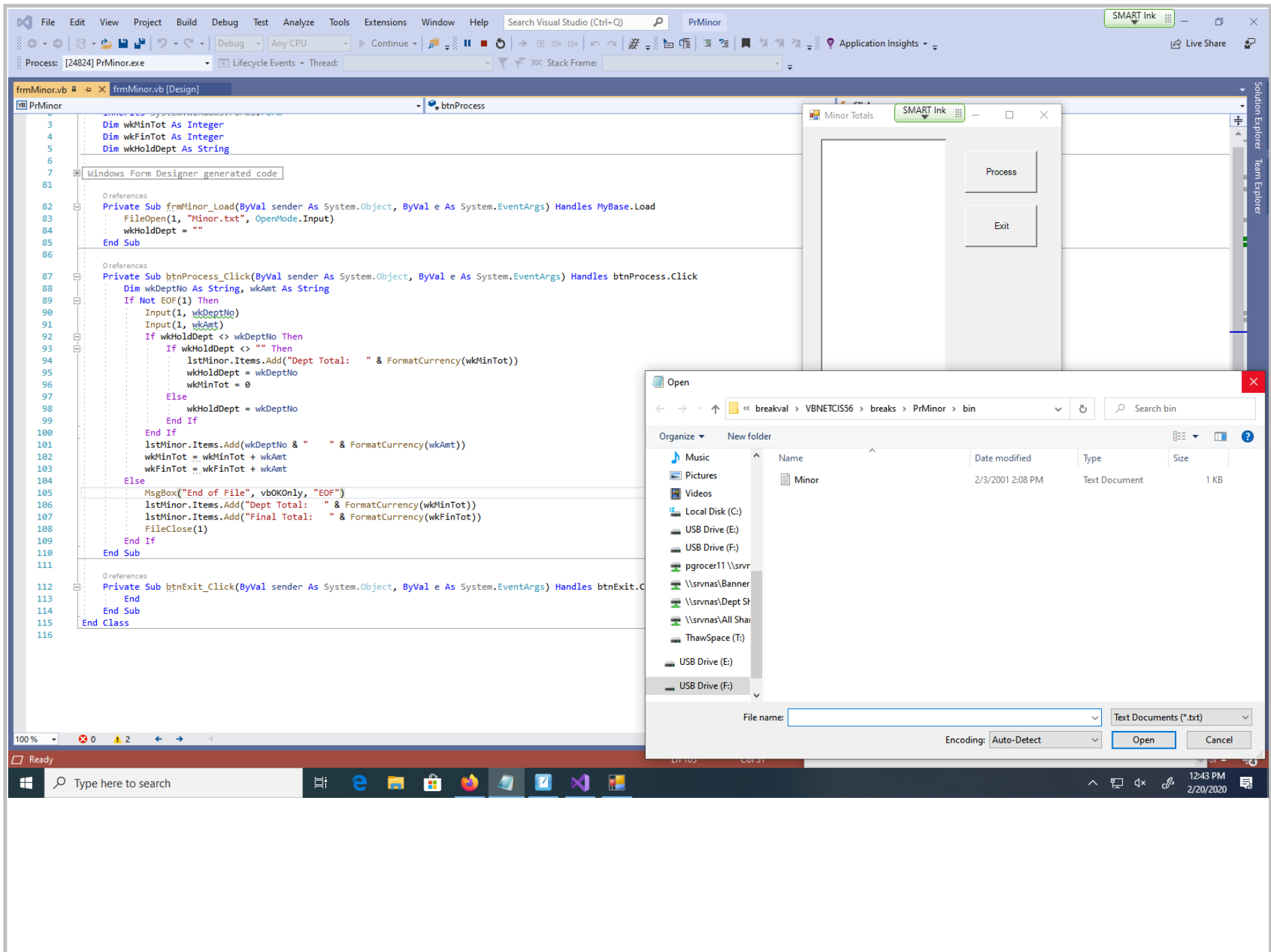
Slide 9 of 9

Type here to search

Comments

113%

12:39 PM
2/20/2020



Visual Studio interface showing the code for PrMinor.vb and its execution results.

```

3 Dim wkMinTot As Integer
4 Dim wkFinTot As Integer
5 Dim wkHoldDept As String
6
7 Windows Form Designer generated code
81
82 Private Sub frmMinor_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
83     FileOpen(1, "Minor.txt", OpenMode.Input)
84     wkHoldDept = ""
85 End Sub
86
87 Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
88     Dim wkDeptNo As String, wkAmt As String
89     If Not EOF(1) Then
90         Input(1, wkDeptNo)
91         Input(1, wkAmt)
92         If wkHoldDept <> wkDeptNo Then
93             If wkHoldDept <> "" Then
94                 lstMinor.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
95                 wkHoldDept = wkDeptNo
96                 wkMinTot = 0
97             Else
98                 wkHoldDept = wkDeptNo
99             End If
100         End If
101         lstMinor.Items.Add(wkDeptNo & " " & FormatCurrency(wkAmt))
102         wkMinTot = wkMinTot + wkAmt
103         wkFinTot = wkFinTot + wkAmt
104     Else
105         MsgBox("End of File", vbOKOnly, "EOF")
106         lstMinor.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
107         lstMinor.Items.Add("Final Total: " & FormatCurrency(wkFinTot))
108         FileClose(1)
109     End If
110 End Sub
111
112 Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
113 End
114 End Sub
115 End Class
116

```

Handwritten annotations in blue and green ink:

- wkHoldDept* (with "||" and "12" below it)
- first rec break* (with "break" below it)
- wkMinTot* (with "500", "600", and "650" below it)
- wkFinTot* (with "500", "600", and "650" below it)

Minor Totals window:

```

12 $500.00
12 $100.00
12 $50.00

```

Minor - Notepad window:

```

"12", "500"
"12", "100"
"12", "050"
"15", "200"
"15", "100"
"15", "300"
"15", "060"
"17", "020"
"17", "240"

```

Visual Studio screenshot showing a VB.NET application named PrMinor. The code in frmMinor.vb includes variables for wkMinTot, wkFinTot, and wkHoldDept. The btnProcess_Click event handler processes departmental data, adding items to a list and updating totals. Handwritten blue annotations include "first rec break" and "breakz" pointing to the initial data entry logic. Red handwritten notes show calculations for wkMinTot (500, 600, 650, 200) and wkFinTot (500, 600, 650, 200). A SMART Ink window titled "Minor Totals" displays a list of entries and a total of \$650.00. A Notepad window shows the raw data entries, with a red arrow pointing to the entry "15", "200".

```

3 Dim wkMinTot As Integer
4 Dim wkFinTot As Integer
5 Dim wkHoldDept As String
6
7 'Windows Form Designer generated code
81
82 Private Sub frmMinor_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
83     FileOpen(1, "Minor.txt", OpenMode.Input)
84     wkHoldDept = ""
85 End Sub
86
87 Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
88     Dim wkDeptNo As String, wkAmt As String
89     If Not EOF(1) Then
90         Input(1, wkDeptNo)
91         Input(1, wkAmt)
92         If wkHoldDept <> wkDeptNo Then
93             If wkHoldDept <> "" Then
94                 lstMinor.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
95                 wkHoldDept = wkDeptNo
96                 wkMinTot = 0
97             Else
98                 wkHoldDept = wkDeptNo
99             End If
100         End If
101         lstMinor.Items.Add(wkDeptNo & " " & FormatCurrency(wkAmt))
102         wkMinTot = wkMinTot + wkAmt
103         wkFinTot = wkFinTot + wkAmt
104     Else
105         MsgBox("End of File", vbOKOnly, "EOF")
106         lstMinor.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
107         lstMinor.Items.Add("Final Total: " & FormatCurrency(wkFinTot))
108         FileClose(1)
109     End If
110 End Sub
111
112 Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
113     End
114 End Sub
115 End Class
116

```

Minor Totals

12	\$500.00
12	\$100.00
12	\$50.00
Total: \$650.00	
15	\$200.00

Minor - Notepad

```

"12", "500"
"12", "100"
"12", "050"
"15", "200"
"15", "100"
"15", "300"
"15", "060"
"17", "020"
"17", "240"

```


The screenshot displays a Visual Studio IDE with the following components:

- Code Editor:** Shows the VB.NET code for the `PrMinor` class. The `btnProcess_Click` method reads a file named `Minor.txt` and processes the data. It uses `Input` to get department numbers and amounts, and `lstMinor.Items.Add` to store the results. A `MsgBox` is shown for "End of File".
- Minor Totals Window:** A window titled "Minor Totals" containing a list of departments and their totals. The list is as follows:

12	\$500.00
12	\$100.00
12	\$50.00
Dept Total:	\$650.00
15	\$200.00
15	\$100.00
15	\$300.00
15	\$60.00
Dept Total:	\$660.00
17	\$20.00
17	\$240.00

Buttons for "Process" and "Exit" are visible on the right.
- Minor - Notepad Window:** A window titled "Minor - Notepad" showing the raw data from the file. The data is as follows:

"12"	"500"
"12"	"100"
"12"	"050"
"15"	"200"
"15"	"100"
"15"	"300"
"15"	"060"
"17"	"020"
"17"	"240"

A red arrow points to the third row of data.
- EOF Dialog Box:** A small dialog box titled "EOF" with the text "End of File" and an "OK" button.

Visual Studio screenshot showing a VB.NET application named PrMinor. The code in frmMinor.vb [Design] includes:

```

3 Dim wkMinTot As Integer
4 Dim wkFinTot As Integer
5 Dim wkHoldDept As String
6
7 Windows Form Designer generated code
81
82 O references
83 Private Sub frmMinor_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
84     FileOpen(1, "Minor.txt", OpenMode.Input)
85     wkHoldDept = ""
86 End Sub
87
88 O references
89 Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
90     Dim wkDeptNo As String, wkAmt As String
91     If Not EOF(1) Then
92         Input(1, wkDeptNo)
93         Input(1, wkAmt)
94         If wkHoldDept <> wkDeptNo Then
95             If wkHoldDept <> "" Then
96                 lstMinor.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
97                 wkHoldDept = wkDeptNo
98                 wkMinTot = 0
99             Else
100                 wkHoldDept = wkDeptNo
101             End If
102             lstMinor.Items.Add(wkDeptNo & " " & FormatCurrency(wkAmt))
103             wkMinTot = wkMinTot + wkAmt
104             wkFinTot = wkFinTot + wkAmt
105         Else
106             MsgBox("End of File", vbOKOnly, "EOF")
107             lstMinor.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
108             lstMinor.Items.Add("Final Total: " & FormatCurrency(wkFinTot))
109             FileClose(1)
110         End If
111     End Sub
112
113 O references
114 Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click
115     End
116 End Sub
117 End Class

```

The application is running, displaying a window titled "Minor Totals" with a list of entries and two buttons: "Process" and "Exit".

Dept No	Amount
12	\$500.00
12	\$100.00
12	\$50.00
Dept Total: \$650.00	
15	\$200.00
15	\$100.00
15	\$300.00
15	\$60.00
Dept Total: \$660.00	
17	\$20.00
17	\$240.00
Dept Total: \$260.00	
Final Total: \$1,570.00	

The Notepad window shows the raw input data from the text file:

```

"12", "500"
"12", "100"
"12", "050"
"15", "200"
"15", "100"
"15", "300"
"15", "060"
"17", "020"
"17", "240"

```

MIMbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

1 Minor, Intermediate and Major Breaks
Please use speaker notes for additional information!

2

3

4

5

6

Slide 2 of 18

Comments

Type here to search

12:56 PM 2/20/2020

Input data for major.cbl

DW Major
Br Intermediate
Dept Minor

Input data: *item #*
item name

0110121234	HOE	01201899
0110121235	RAKE	02501595
0110151111	TROWEL	02000599
0110151112	HAND HOE	01400799
0120171000	LILY	05000199
0120171001	TULIP	07500299
0121223000	PANSY	11200599
0121223001	PETUNIA	10000499
0230245555	DOGWOOD	01010099
0230245556	CHERRY	01012000
0240276789	WHITE PINE	00513500
0240276790	BLACK PINE	01020000
0250553456	MAPLE	00807500
0250553457	OAK	00909900

— qual
— price

This is the input data for the program called major.cbl that is available at the site. It is the one that was also discussed in class.

MIMbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

1 Minor, Intermediate and Major Breaks
Please use speaker notes for additional information!

2

3

4

5

6

First 7 records:

The first seven input records:

```

0110121234HOE
0110121235RAKE
0110151111TROWEL
0110151112HAND HOE
0120171000LILY
0120171001TULIP
0121223000PANSY
          
```

04/11/99

		INVENTORY REPORT				PAGE 1		
DIV	BR	DEPT	ITEM #	ITEM NAME	ON HAND	PRICE	INVEN VALUE	
01	10	12	1234	HOE	12	\$18.99	\$227.88	
01	10	12	1235	RAKE	25	\$15.95	\$398.75	
TOTALS FOR DEPT: 12					37		\$626.63	
01	10	15	1111	TROWEL	20	\$5.99	\$119.80	
01	10	15	1112	HAND HOE	14	\$7.99	\$111.86	
TOTALS FOR DEPT: 15					34		\$231.66	
TOTALS FOR BRANCH: 10					71		\$858.29	
01	20	17	1000	LILY	50	\$1.99	\$99.50	
01	20	17	1001	TULIP	75	\$2.99	\$224.25	
TOTALS FOR DEPT: 17					125		\$323.75	
TOTALS FOR BRANCH: 20					125		\$323.75	
01	21	22	3000	PANSY	112	\$5.99	\$670.88	

This shows the first seven records and the resulting report.
Notice then when the dept breaks, I print a dept total before processing the record that caused the break and when the br changes, I print a branch total before processing the record that caused the break.

Note that when the dept changes from 12 to 15, a total is written for dept 12 before dept 15 is processed.

Note when the BR changes from 10 to 20, I need to write a dept total for the last dept and then a br total for the br before I process the first record in br 20.

MIMbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

1 Minor, Intermediate and Major Breaks
Please use speaker notes for additional information!

2

3

4

5

6

```

graph TD
    MAINLINE([MAINLINE]) --> A100[A-100-INITIALIZE]
    A100 --> B100[B-100-PROCESS]
    B100 --> C100[C-100-WRAPUP]
    C100 --> STOP([STOP RUN])
    STOP --> A100_2[A-100-INITIALIZE]
    A100_2 --> OPEN[OPEN FILES]
    OPEN --> DATE[PROCESS DATE]
    DATE --> END_A([END A-100])
    
    B100 --> READ[/READ 1ST RECORD/]
    READ -- AT END --> NO_MORE[NO TO MORE-RECS]
    NO_MORE -.-> READ
    
    READ --> HOLD[DIV TO DIV-HOLD  
BR TO BR-HOLD  
DEPT TO DEPT-HOLD]
    HOLD --> DEC{MORE-RECS  
NOT = NO}
    DEC -- Y --> LOOP[B-200-LOOP]
    LOOP --> HOLD
    DEC -- N --> MINOR[B-310-MINOR-ROUT]
    MINOR --> INTER[B-320-INTER-ROUT]
    INTER --> A_CONN((A))
    A_CONN --> MAJOR[B-330-MAJOR-ROUT]
    MAJOR --> FINAL[B-340-FINAL-ROUT]
    FINAL --> END_B([END B-100])
    
```

NOTE: DIV is the major break field, BR is the intermediate break field, and DEPT is the minor break field in this example. The move to hold should move whatever is the major break field, the intermediate break field and the minor break field.

This is the logic flowchart - page 1.

Slide 5 of 18

Type here to search

Comments

113%

1:01 PM
2/20/2020

MIMbreak [Protected View] - PowerPoint

File Home Insert Draw Design Transitions Animations Slide Show Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

3

4

5

6

7

8

```

graph TD
    Start([B-200-LOOP]) --> Div{DIV NOT = DIV-HOLD}
    Div -- N --> Br{BR NOT = BR-HOLD}
    Div -- Y --> B310_1[B-310-MINOR-ROUT]
    Br -- N --> Dept{DEPT NOT = DEPT-HOLD}
    Br -- Y --> B310_2[B-310-MINOR-ROUT]
    Dept -- N --> Br
    Dept -- Y --> B310_3[B-310-MINOR-ROUT]
    B310_1 -- N --> B230_1[B-230-INTER-ROUT]
    B310_1 -- Y --> B230_2[B-230-INTER-ROUT]
    B310_2 -- N --> B230_1
    B310_2 -- Y --> B230_2
    B230_1 -- N --> B240[B-240-MAJOR-ROUT]
    B230_1 -- Y --> B240
    B230_2 -- N --> B240
    B230_2 -- Y --> B240
    B240 --> Conn1(( ))
    B310_3 --> Conn2(( ))
    Conn1 --> Conn2
    Conn2 --> End[B]
  
```

This is the logic flowchart - page 2.
 Note that I check for the highest level of break first because I am making the assumption that if the division changes, the branch and department also change. If this is not a valid assumption, I would need to code differently.

Slide 6 of 18

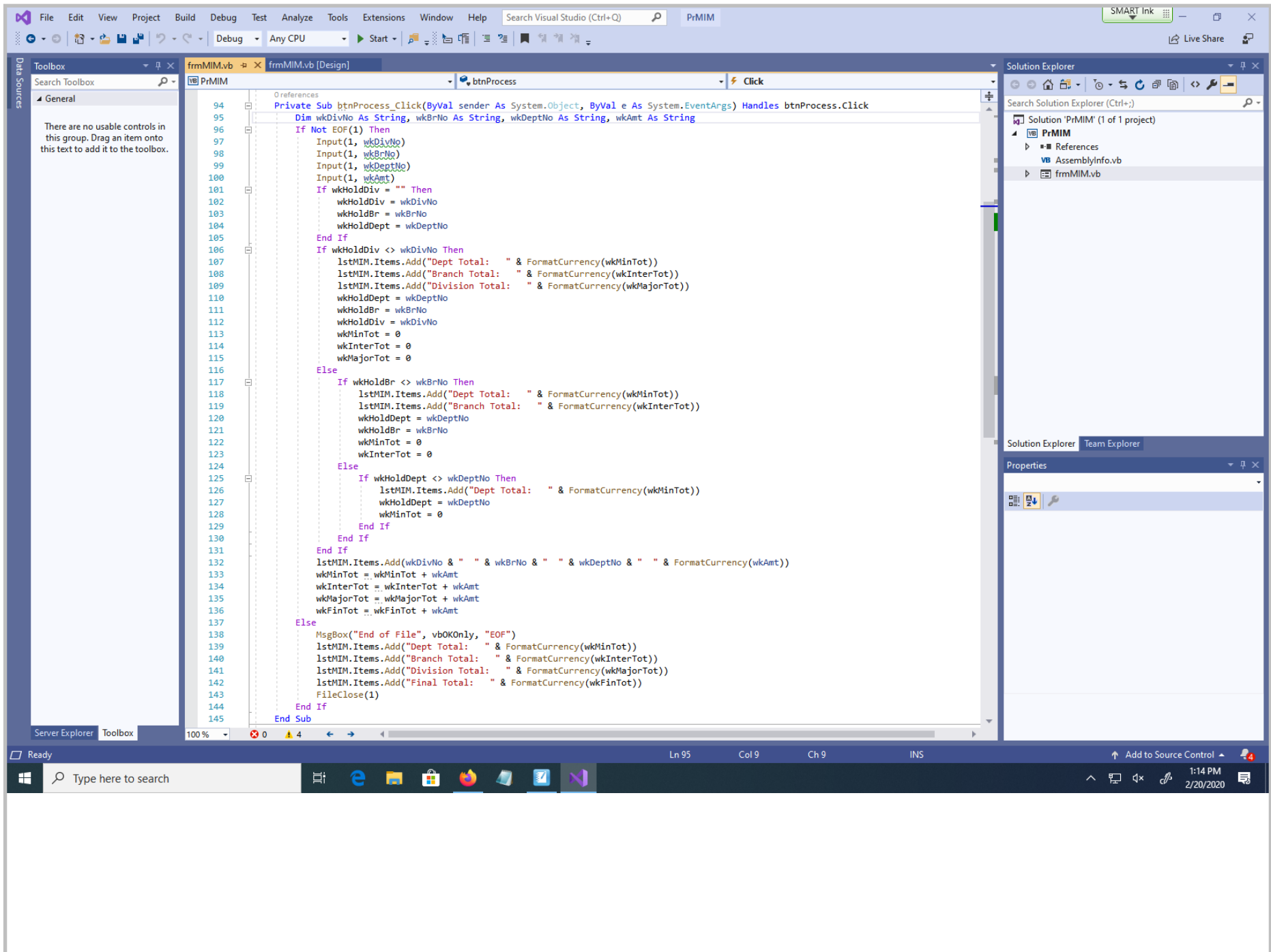
Comments

Type here to search

1:04 PM 2/20/2020

Visual Studio interface showing the code for `frmMIM.vb`. The code is in VB.NET and includes two event handlers: `frmMinor_Load` and `btnProcess_Click`.

```
11 Windows Form Designer generated code
12
13 0 references
14 Private Sub frmMinor_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
15     FileOpen(1, "MIM.txt", OpenMode.Input)
16     wkHoldDept = ""
17     wkHoldBr = ""
18     wkHoldDiv = ""
19 End Sub
20
21 0 references
22 Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
23     Dim wkDivNo As String, wkBrNo As String, wkDeptNo As String, wkAmt As String
24     If Not EOF(1) Then
25         Input(1, wkDivNo)
26         Input(1, wkBrNo)
27         Input(1, wkDeptNo)
28         Input(1, wkAmt)
29         If wkHoldDiv = "" Then
30             wkHoldDiv = wkDivNo
31             wkHoldBr = wkBrNo
32             wkHoldDept = wkDeptNo
33         End If
34         If wkHoldDiv <> wkDivNo Then
35             lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
36             lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
37             lstMIM.Items.Add("Division Total: " & FormatCurrency(wkMajorTot))
38             wkHoldDept = wkDeptNo
39             wkHoldBr = wkBrNo
40             wkHoldDiv = wkDivNo
41             wkMinTot = 0
42             wkInterTot = 0
43             wkMajorTot = 0
44         Else
45             If wkHoldBr <> wkBrNo Then
46                 lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
47                 lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
48                 wkHoldDept = wkDeptNo
49                 wkHoldBr = wkBrNo
50                 wkMinTot = 0
51                 wkInterTot = 0
52             Else
53                 If wkHoldDept <> wkDeptNo Then
54                     lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
55                     wkHoldDept = wkDeptNo
56                     wkMinTot = 0
57                 End If
58             End If
59         End If
60         lstMIM.Items.Add(wkDivNo & " " & wkBrNo & " " & wkDeptNo & " " & FormatCurrency(wkAmt))
61         wkMinTot = wkMinTot + wkAmt
62         wkInterTot = wkInterTot + wkAmt
63         wkMajorTot = wkMajorTot + wkAmt
64     End If
65 End Sub
```



The screenshot displays the Visual Studio IDE with a VB.NET code file open. The code is a private sub procedure named `btnProcess_Click` that handles the click of a button. It prompts the user for several inputs: `wkDivNo`, `wkBrNo`, `wkDeptNo`, and `wkAmt`. Based on these inputs, it calculates various totals (Min, Inter, Major, Fin) and adds them to a list named `lstMIM`. The code also includes logic to handle the end of a file and to display a message box.

Overlaid on the code editor is a Notepad window titled "MIM - Notepad". It contains a list of data entries, each consisting of four comma-separated values in quotes, representing the output of the application. The entries are as follows:

```
"03", "24", "27", "500"  
"03", "24", "27", "600"  
"03", "24", "27", "200"  
"03", "24", "28", "150"  
"03", "24", "28", "275"  
"03", "24", "28", "620"  
"03", "25", "15", "120"  
"03", "25", "15", "175"  
"03", "25", "15", "600"  
"03", "25", "17", "500"  
"03", "25", "17", "100"  
"03", "25", "20", "150"  
"03", "25", "20", "220"  
"04", "24", "27", "125"  
"04", "24", "27", "250"  
"04", "24", "27", "450"  
"04", "24", "27", "600"  
"04", "24", "28", "100"  
"04", "24", "28", "126"  
"04", "24", "29", "600"  
"04", "24", "29", "240"  
"04", "25", "15", "120"  
"04", "25", "17", "600"  
"04", "25", "17", "555"
```

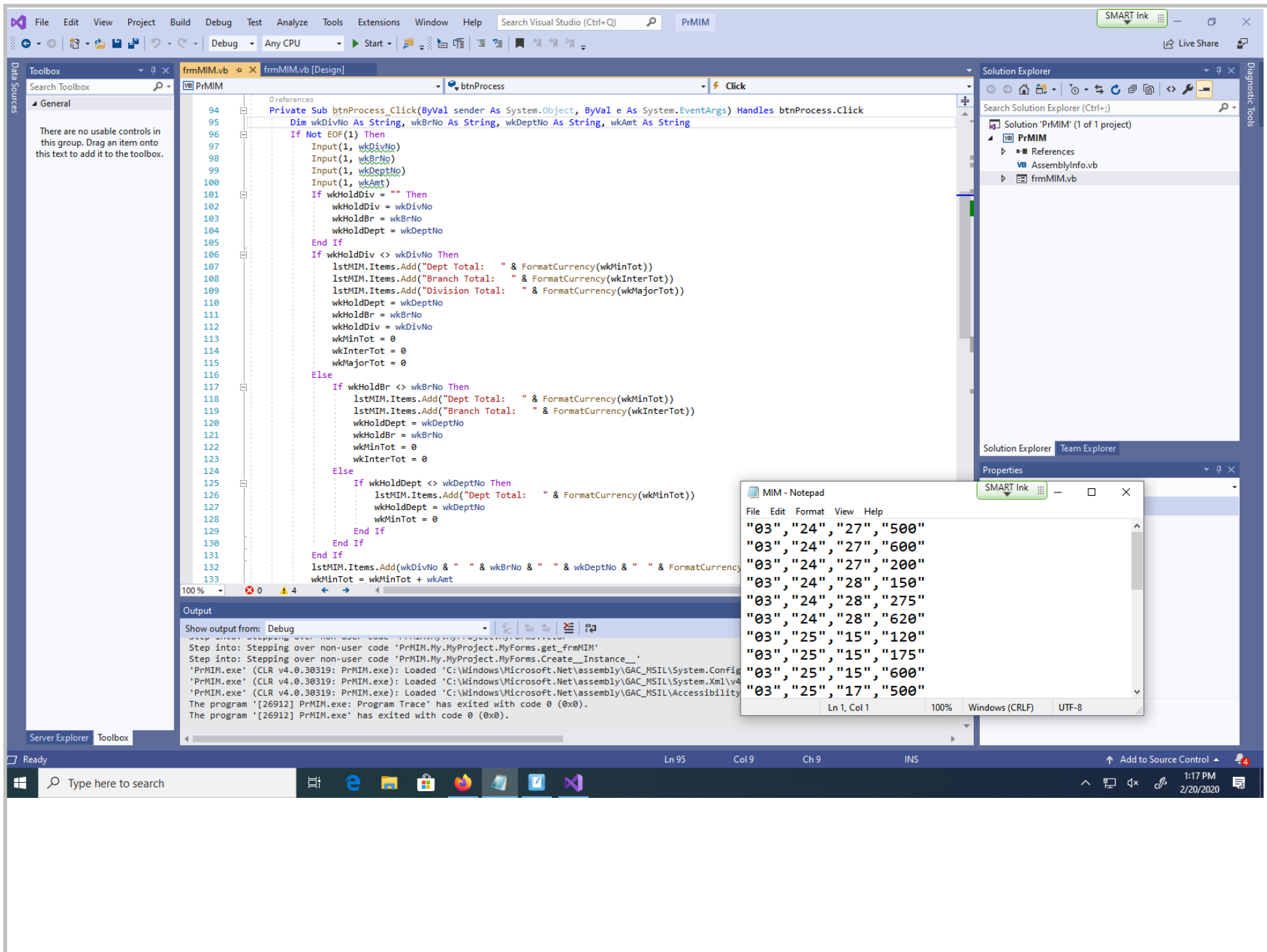
The screenshot displays the Visual Studio IDE with the following components:

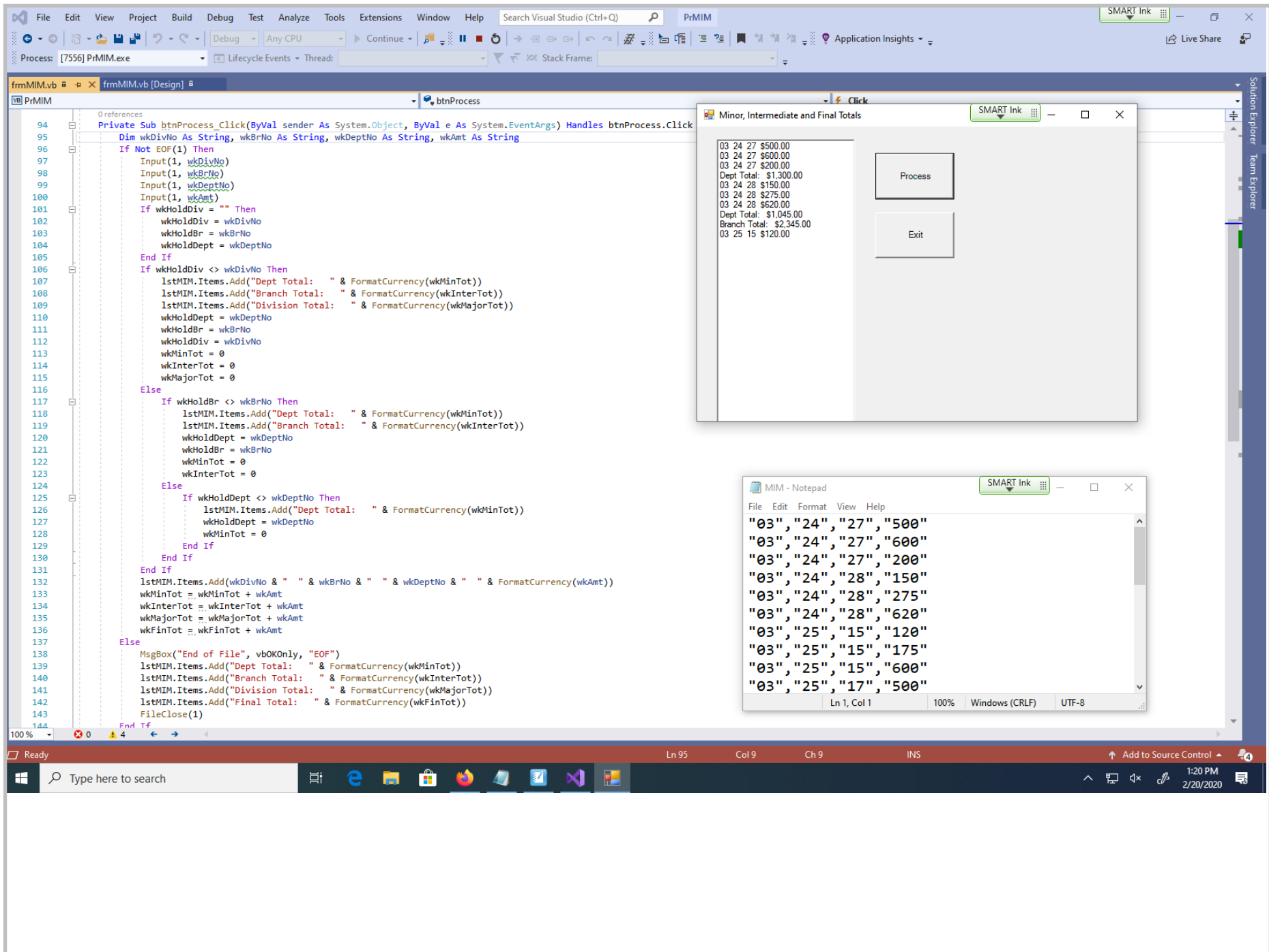
- Code Editor:** Shows a VB.NET subroutine `btnProcess_Click` in `frmMIM.vb`. The code processes input data and calculates totals for departments, branches, and divisions. It includes logic for handling EOF (End of File) and updating a list of items (`lstMIM.Items`).
- Minor, Intermediate and Final Totals:** A dialog box with a text area containing the input data: `03 24 27 $500.00`. It features two buttons: `Process` and `Exit`.
- MIM - Notepad:** A text editor window displaying the output of the application, showing a list of formatted strings representing the calculated totals for various departments and branches.

```
Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
    Dim wkDivNo As String, wkBrNo As String, wkDeptNo As String, wkAmt As String
    If Not EOF(1) Then
        Input(1, wkDivNo)
        Input(1, wkBrNo)
        Input(1, wkDeptNo)
        Input(1, wkAmt)
        If wkHoldDiv = "" Then
            wkHoldDiv = wkDivNo
            wkHoldBr = wkBrNo
            wkHoldDept = wkDeptNo
        End If
        If wkHoldDiv <> wkDivNo Then
            lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
            lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
            lstMIM.Items.Add("Division Total: " & FormatCurrency(wkMajorTot))
            wkHoldDept = wkDeptNo
            wkHoldBr = wkBrNo
            wkHoldDiv = wkDivNo
            wkMinTot = 0
            wkInterTot = 0
            wkMajorTot = 0
        Else
            If wkHoldBr <> wkBrNo Then
                lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
                lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
                wkHoldDept = wkDeptNo
                wkHoldBr = wkBrNo
                wkMinTot = 0
                wkInterTot = 0
            Else
                If wkHoldDept <> wkDeptNo Then
                    lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
                    wkHoldDept = wkDeptNo
                    wkMinTot = 0
                End If
            End If
        End If
        lstMIM.Items.Add(wkDivNo & " " & wkBrNo & " " & wkDeptNo & " " & FormatCurrency(wkAmt))
        wkMinTot = wkMinTot + wkAmt
        wkInterTot = wkInterTot + wkAmt
        wkMajorTot = wkMajorTot + wkAmt
        wkFinTot = wkFinTot + wkAmt
    Else
        MsgBox("End of File", vbOKOnly, "EOF")
        lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
        lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
        lstMIM.Items.Add("Division Total: " & FormatCurrency(wkMajorTot))
        lstMIM.Items.Add("Final Total: " & FormatCurrency(wkFinTot))
        FileClose(1)
    End If
End Sub
```

MIM - Notepad output:

```
"03", "24", "27", "500"  
"03", "24", "27", "600"  
"03", "24", "27", "200"  
"03", "24", "28", "150"  
"03", "24", "28", "275"  
"03", "24", "28", "620"  
"03", "25", "15", "120"  
"03", "25", "15", "175"  
"03", "25", "15", "600"  
"03", "25", "17", "500"
```





File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search Visual Studio (Ctrl+Q) PrMIM SMART Ink Live Share

Process: [29040] PrMIM.exe Lifecycle Events Thread: [24080] Main Thread Stack Frame: PrMIM.frmMIM.btnProcess_Click

frmMIM.vb x frmMIM.vb [Design] PrMIM btnProcess Click

```

94 Private Sub btnProcess_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProcess.Click
95     Dim wkDivNo As String, wkBrNo As String, wkDeptNo As String, wkAmt As String
96     If Not EOF(1) Then
97         Input(1, wkDivNo)
98         Input(1, wkBrNo)
99         Input(1, wkDeptNo)
100        Input(1, wkAmt)
101        If wkHoldDiv = "" Then
102            wkHoldDiv = wkDivNo
103            wkHoldBr = wkBrNo
104            wkHoldDept = wkDeptNo
105        End If
106        If wkHoldDiv <> wkDivNo Then
107            lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
108            lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
109            lstMIM.Items.Add("Division Total: " & FormatCurrency(wkMajorTot))
110            wkHoldDept = wkDeptNo
111            wkHoldBr = wkBrNo
112            wkHoldDiv = wkDivNo
113            wkMinTot = 0
114            wkInterTot = 0
115            wkMajorTot = 0
116        Else
117            If wkHoldBr <> wkBrNo Then
118                lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
119                lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
120                wkHoldDept = wkDeptNo
121                wkHoldBr = wkBrNo
122                wkMinTot = 0
123                wkInterTot = 0
124            Else
125                If wkHoldDept <> wkDeptNo Then
126                    lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
127                    wkHoldDept = wkDeptNo
128                    wkMinTot = 0
129                End If
130            End If
131        End If
132        lstMIM.Items.Add(wkDivNo & " " & wkBrNo & " " & wkDeptNo & " " & FormatCurrency(wkAmt))
133        wkMinTot = wkMinTot + wkAmt
134        wkInterTot = wkInterTot + wkAmt
135        wkMajorTot = wkMajorTot + wkAmt
136        wkFinTot = wkFinTot + wkAmt
137    Else
138        MsgBox("End of File", vbOKOnly, "EOF")
139        lstMIM.Items.Add("Dept Total: " & FormatCurrency(wkMinTot))
140        lstMIM.Items.Add("Branch Total: " & FormatCurrency(wkInterTot))
141        lstMIM.Items.Add("Division Total: " & FormatCurrency(wkMajorTot))
142        lstMIM.Items.Add("Final Total: " & FormatCurrency(wkFinTot))
143        FileClose(1)
144    End If

```

Minor, Intermediate and Final Totals

SMART Ink

Process

Exit

```

03 25 15 $600.00
Dept Total: $995.00
03 25 17 $500.00
03 25 17 $100.00
Dept Total: $600.00
03 25 20 $150.00
03 25 20 $220.00
Dept Total: $370.00
Branch Total: $1,865.00
Division Total: $4,210.00
04 24 27 $125.00
04 24 27 $250.00
04 24 27 $450.00
04 24 27 $600.00
Dept Total: $1,425.00
04 24 28 $100.00
04 24 28 $126.00
Dept Total: $226.00
04 24 29 $600.00
04 24 29 $240.00
Dept Total: $840.00
Branch Total: $2,491.00
04 25 15 $120.00
Dept Total: $120.00
04 25 17 $600.00
04 25 17 $555.00
Dept Total: $1,155.00
Branch Total: $1,275.00
Division Total: $3,766.00
Final Total: $7,976.00

```

100% Ready Ln 96 Col 9 Ch 9 INS Add to Source Control 1:21 PM 2/20/2020

Visual Studio interface showing a VB.NET code file named `PrMIM` and a running application window titled "Minor, Intermediate and Final Totals".

The code in `PrMIM` is a `Private Sub btnProcess_Click` method that processes data from a file. It uses a list `lstMIM` to store formatted data and calculates various totals (Dept, Branch, Division, Major, Final) based on the input data. The code includes logic for handling "EOF" (End of File) and displaying a message box.

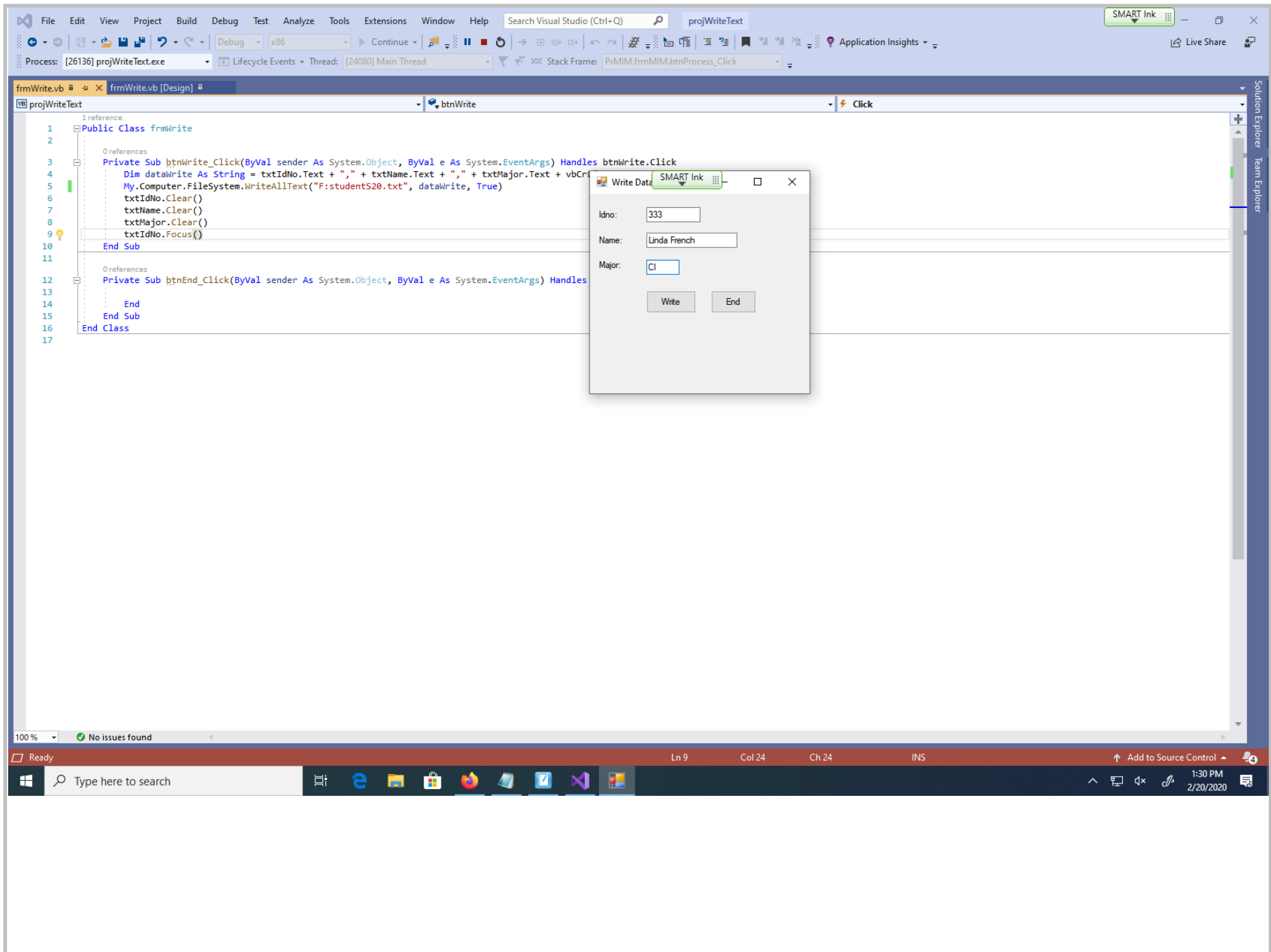
The application window displays the following data:

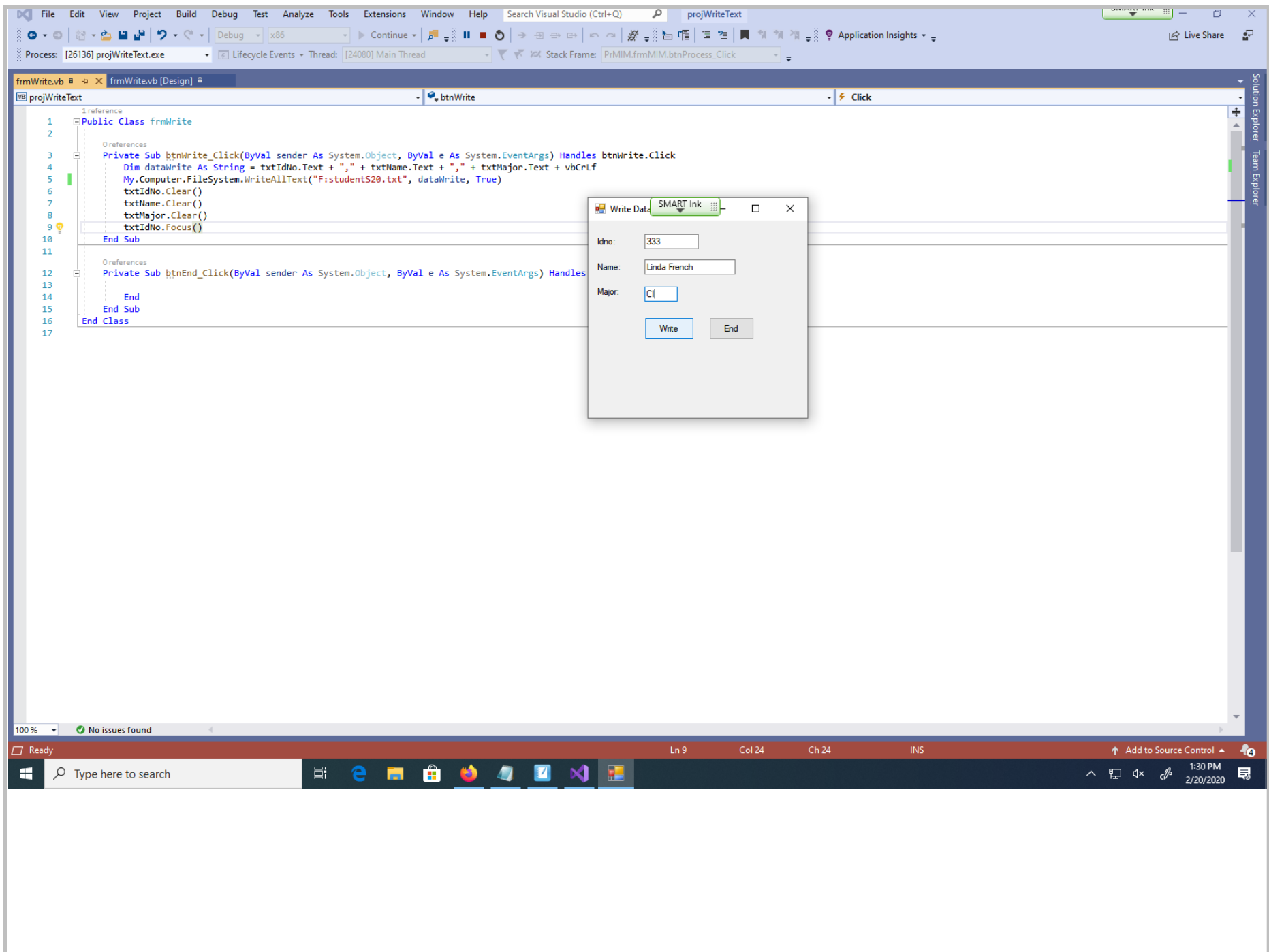
```

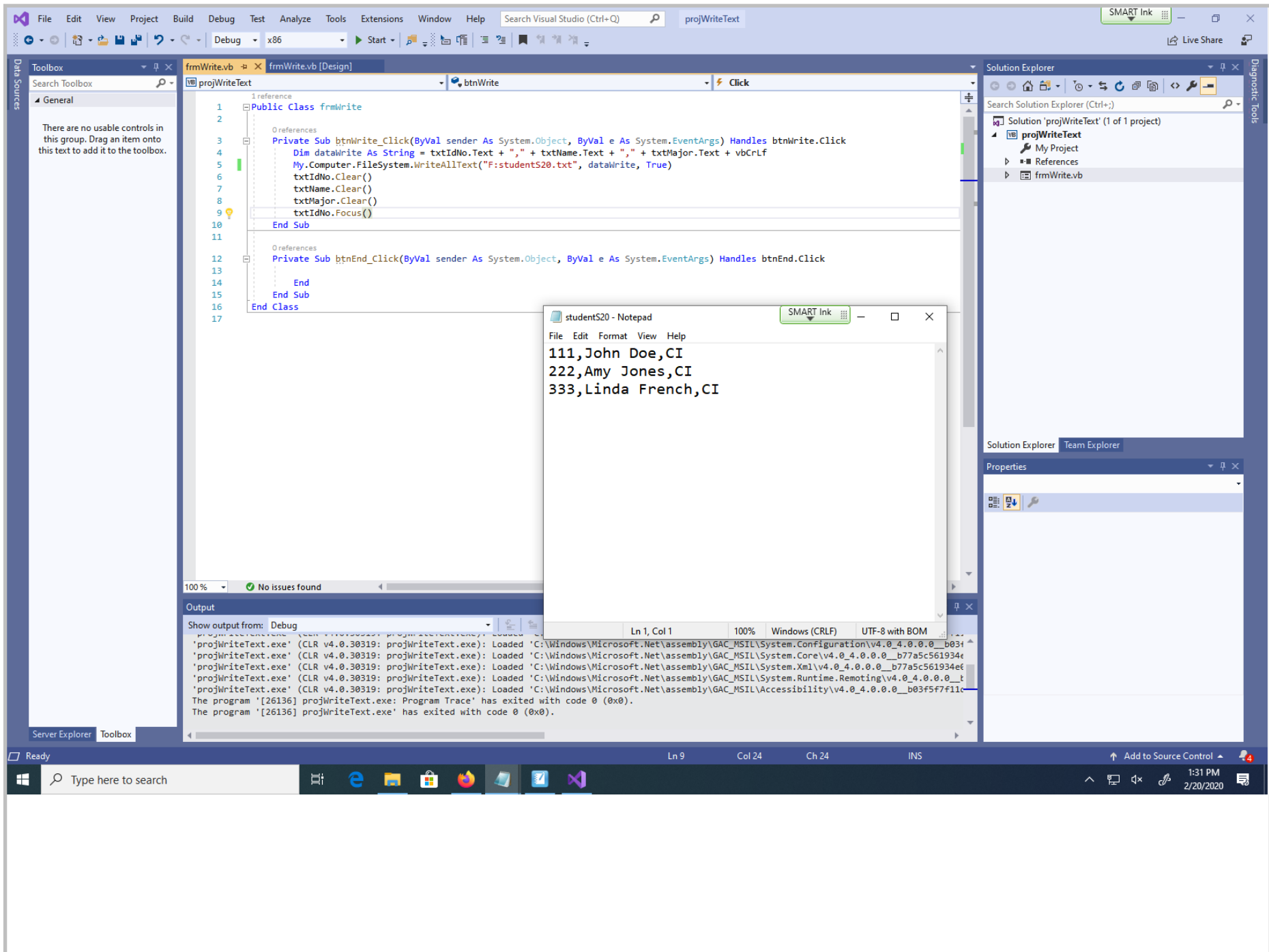
03 25 15 $600.00
Dept Total: $895.00
03 25 17 $500.00
03 25 17 $100.00
Dept Total: $600.00
03 25 20 $150.00
03 25 20 $220.00
Dept Total: $370.00
Branch Total: $1,865.00
Division Total: $4,210.00
04 24 27 $125.00
04 24 27 $250.00
04 24 27 $450.00
04 24 27 $600.00
Dept Total: $1,425.00
04 24 28 $100.00
04 24 28 $126.00
Dept Total: $226.00
04 24 29 $600.00
04 24 29 $240.00
Dept Total: $840.00
Branch Total: $2,491.00
04 25 15 $120.00
Dept Total: $120.00
04 25 17 $600.00
04 25 17 $555.00
Dept Total: $1,155.00
Branch Total: $1,275.00
Division Total: $3,766.00
Final Total: $7,976.00

```

The application window also features a "Process" button and an "Exit" button.





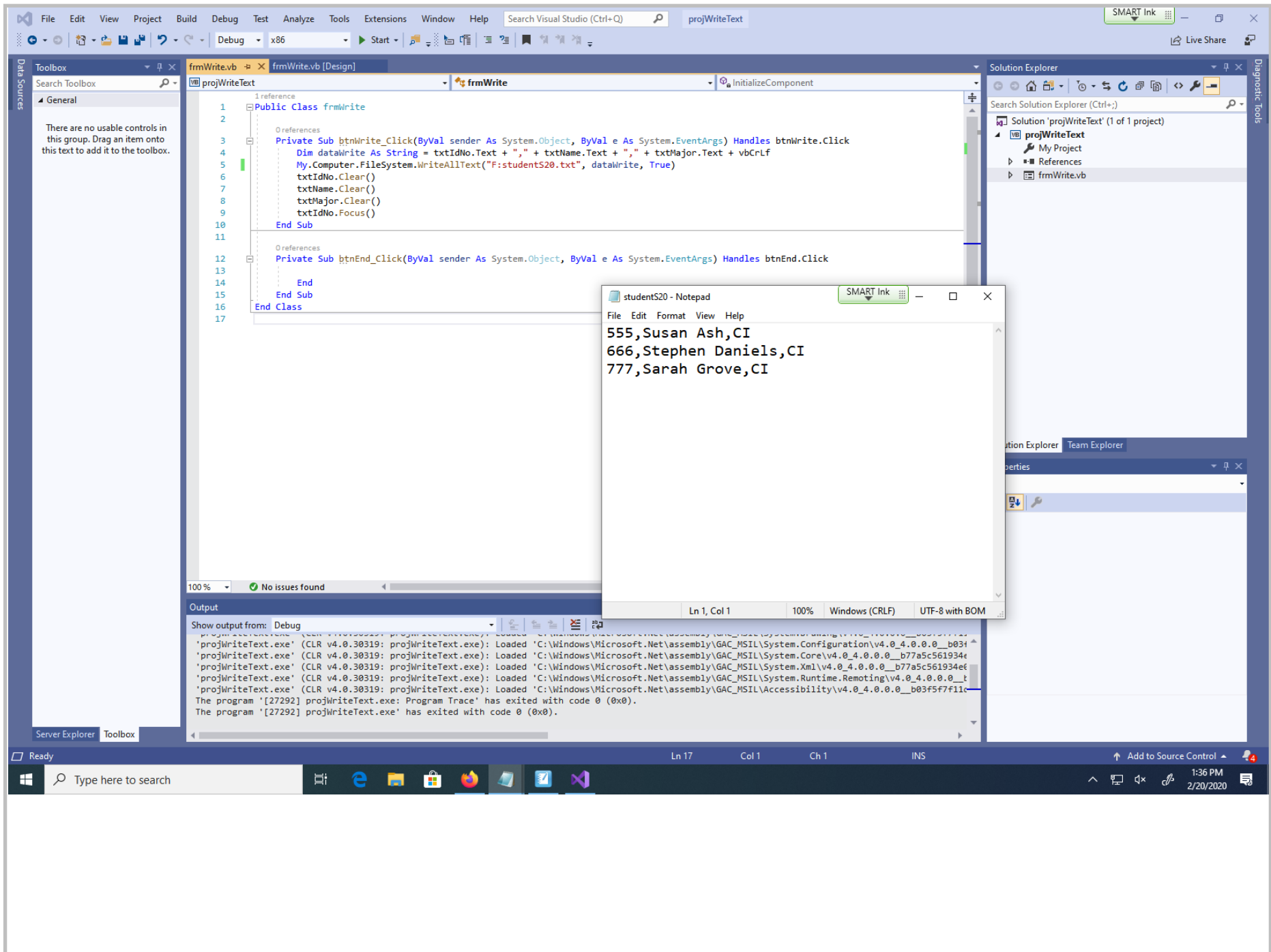


The image shows a screenshot of the Visual Studio IDE. The main window displays the source code for a class named `frmWrite`. The code includes a public class declaration, a private event handler `btnWrite_Click` that writes data to a file, and another private event handler `btnEnd_Click`. A dialog box titled "Write Data" is overlaid on the code, featuring three text input fields labeled "Idno:", "Name:", and "Major:", and two buttons labeled "Write" and "End".

```
1 reference
2
3 Public Class frmWrite
4     Private Sub btnWrite_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnWrite.Click
5         Dim dataWrite As String = txtIdNo.Text + "," + txtName.Text + "," + txtMajor.Text + vbCrLf
6         My.Computer.FileSystem.WriteAllText("F:\students20.txt", dataWrite, False)
7         txtIdNo.Clear()
8         txtName.Clear()
9         txtMajor.Clear()
10        txtIdNo.Focus()
11    End Sub
12
13    Private Sub btnEnd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEnd.Click
14        End
15    End Sub
16 End Class
17
```

The image shows a Visual Studio IDE window with the following components:

- Code Editor:** Displays the code for `frmWrite`. It includes a `Public Class frmWrite` with two event handlers: `btnWrite_Click` and `btnEnd_Click`. The `btnWrite_Click` method concatenates text from `txtIdNo`, `txtName`, and `txtMajor` into `dataWrite` and calls `My.Computer.FileSystem.WriteAllText` with `False` as the second parameter. The `btnEnd_Click` method is currently empty.
- Notepad Window:** A window titled `studentS20 - Notepad` is open, displaying the text `555,Susan Ash,CI`. This text corresponds to the output of the `btnWrite_Click` method.
- Output Window:** Shows the debug output for `projWriteText.exe`, including assembly loading information and a program trace indicating that the program exited with code `0 (0x0)`.
- UI Elements:** The Visual Studio interface includes a menu bar, a toolbar, a Solution Explorer on the right showing the project structure, and a Properties window at the bottom right. A blue annotation is present in the code editor: "True adds to what is there - false overwrites with new data." pointing to the `False` parameter in the `WriteAllText` call.



The image shows a screenshot of the Visual Studio IDE. The main window displays the code for a VB.NET application named 'readProj'. The code defines a class 'frmRead' with a 'btnRead' button. The 'btnRead_Click' event handler reads a file 'F:\studentS20.txt' and displays the first row of data in text boxes. The 'frmRead_Load' event handler sets the 'MyReader' to read a delimited file.

```
2 Dim MyReader As New Microsoft.VisualBasic.FileIO.TextFieldParser("F:\studentS20.txt")
3
4
5 Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRead.Click
6
7     Dim currentRow As String()
8     If Not MyReader.EndOfData Then
9
10        currentRow = MyReader.ReadFields()
11        txtIdNo.Text = currentRow(0)
12        txtName.Text = currentRow(1)
13        txtMajor.Text = currentRow(2)
14
15    Else
16        Label1.Text = "End of file reached!"
17    End If
18 End Sub
19
20 Private Sub frmRead_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
21
22     MyReader.TextFieldType = FileIO.FieldType.Delimited
23     MyReader.SetDelimiters(",")
24
25 End Sub
26 End Class
27
```

Overlaid on the code is a window titled 'Form1' with a 'SMART Ink' logo. It contains three text boxes labeled 'IdNo:', 'Name:', and 'Major:'. The 'IdNo:' box contains '555', the 'Name:' box contains 'Susan Ash', and the 'Major:' box contains 'CI'. A 'Read' button is located below the text boxes.

