

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Data.Odbc;
11 using System.Net.Mail;
12 namespace Avatar_Solutions
13 {
14     public partial class Processor : Form
15     {
16         public Processor()
17         {
18             InitializeComponent();
19         }
20
21         private AvatarConnection avConn;
22         private OdbcConnection conn;
23         private OdbcCommand cmd;
24         private OdbcDataAdapter DA;
25         DataTable dtbl;
26
27         private List<Report> rptList;
28         private List<ServiceRecord> srvRecord;
29
30         int TotalCreated = 0;
31         int TotalError = 0;
32         int TotalAlreadyCreated = 0;
33
34         public Processor(AvatarConnection con)
35         {
36             InitializeComponent();
37             avConn = con;
38             dtbl = new DataTable();
39             conn = avConn.conn;
40
41             srvRecord = new List<ServiceRecord>();
42             rptList = new List<Report>();
43
44             DA = new OdbcDataAdapter();
45
46             dataGridView1.DataSource = dtbl;
47
48
49         }
50
51         private void GetData()
52         {
53             DateTime FromDate = DateTime.Parse(((int)comboBox1.SelectedItem).ToString()
54             + "/01/" + comboBox2.Text);
55             DateTime ThruDate = FromDate.AddMonths(1).AddDays(-1);
56
57             string sendFrom = FromDate.ToString("yyyy-MM-dd");
58             string sendThru = ThruDate.ToString("yyyy-MM-dd");
59             //string sendFrom = FromDate.Value.ToString("yyyy-MM-dd");
60             //string sendThru = ThruDate.Value.ToString("yyyy-MM-dd");
61             //MessageBox.Show("From: " + sendFrom + Environment.NewLine + "Thru: " +
62             sendThru);
63             DA.SelectCommand = avConn.SQLCaseMgmt(sendFrom, sendThru,
64             avConn.Components, avConn.ProgramExclude);
65             DA.SelectCommand.Connection = conn;
66             DA.Fill(dtbl);
67         }
68     }
69 }

```

```

67     private void ProcessData()
68     {
69         int ReqFTF = avConn.ReqFTF;
70         int ReqTot = avConn.ReqTot;
71         string billableCode = avConn.BillableCode;
72
73         //This Code Generates Billable Charges
74         int NonFTF = 0;
75         int FTF = 0;
76         int Total = 0;
77         bool FoundBillable = false;
78         bool started = true;
79         string PreviousClient = "";
80
81
82         int Episode = 0;
83         int PrvEpisode = 0;
84         string currentClient = "";
85         string Service = "";
86         int Tot = 0;
87         string Program = "";
88         string StaffID = "";
89         DateTime DOS = DateTime.Now;
90
91         string PrvService = "";
92         //int Tot = 0;
93         string PrvProgram = "";
94         string PrvStaffID = "";
95         DateTime PrvDOS = DateTime.Now;
96
97         int i = 0;
98         foreach(DataRow row in dtbl.Rows)
99         {
100             i++;
101
102             currentClient = (string)row["PATID"];
103
104
105             if (i == dtbl.Rows.Count || started)
106             {
107                 PreviousClient = currentClient;
108                 started = false;
109
110                 Service = (string)row["Service"];
111                 Tot = (int)row["Tot"];
112                 Program = (string)row["Program"];
113                 StaffID = (string)row["STAFFID"];
114                 Episode = (int)row["EPISODE"];
115                 if (row["DOS"] != DBNull.Value)
116                 {
117                     DOS = Convert.ToDateTime(row["DOS"]);
118                 }
119                 else
120                 {
121                     throw new Exception("No DateTime");
122                 }
123             }
124
125             /*         if (avConn.CurrentRuleTitle == "ACT")
126             {
127                 started = false;
128             }*/
129
130             if (i == dtbl.Rows.Count)
131             {
132                 currentClient = "";
133                 FoundBillable = false;
134                 FTF = 0;
135                 Total = 0;

```

```

136         NonFTF = 0;
137
138         if (Service == billableCode)
139         {
140             FoundBillable = true;
141
142         }
143         else if (Service[Service.Length - 1] == 'F')
144         {
145             FTF += Tot;
146             Total += Tot;
147         }
148         else
149         {
150             NonFTF += Tot;
151             Total += Tot;
152         }
153
154         PrvDOS = DOS;
155         PrvProgram = Program;
156         PrvService = Service;
157         PrvStaffID = StaffID;
158         PrvEpisode = (int)row["EPISODE"];
159
160     }
161
162
163     if (PreviousClient != currentClient)
164     {
165
166         //If override staff logic then change staff id HERE since always
167         //hit.
168         if (avConn.OverrideStaff)
169         {
170             PrvStaffID = avConn.OverrideStaffID;
171         }
172         //end staff override logic
173
174         if (!FoundBillable && FTF >= ReqFTF && Total >= ReqTot)
175         {
176             //GenerateCharge
177             rptList.Add(new Report(DOS, PreviousClient, FTF, Total, true,
178                                     "Charge Created"));
179             avConn.ServList.Add(new ServiceRecord(PrvDOS, PrvEpisode,
180                                                     billableCode, PrvProgram, PrvStaffID, PreviousClient,
181                                                     avConn.currentServiceType));
182             TotalCreated += 1;
183         }
184         else if (FoundBillable)
185         {
186             //WasAlreadyCharges
187             rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false,
188                                     "Already Created"));
189             TotalAlreadyCreated += 1;
190         }
191         else if (FTF >= ReqFTF && Total < ReqTot)
192         {
193             //Not Enough Services
194             rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false,
195                                     "Not Enough Services"));
196             TotalError += 1;
197         }
198         else if (FTF <= ReqFTF)
199         {
200             //Not Enough FTF
201             rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false,
202                                     "Not Enough FTF"));
203             TotalError += 1;
204         }
205         else

```

```

198         {
199             //Other Error
200             rptList.Add(new Report(DOS, PreviousClient, FTF, Total, false,
                "Other Err"));
                TotalError += 1;
            }

            NonFTF = 0;
            FTF = 0;
            Total = 0;
            FoundBillable = false;
        }

        //Always runs -- was miscounted
        Service = (string)row["Service"];
        Tot = (int)row["Tot"];
        Program = (string)row["Program"];
        StaffID = (string)row["STAFFID"];

        if (row["DOS"] != DBNull.Value)
        {
            DOS = Convert.ToDateTime(row["DOS"]);
        }
        else
        {
            throw new Exception("No DateTime");
        }
        if (Service == billableCode)
        {
            FoundBillable = true;
        }
        else if (Service[Service.Length - 1] == 'F')
        {
            FTF += Tot;
            Total += Tot;
        }
        else
        {
            NonFTF += Tot;
            Total += Tot;
        }

        PrvDOS = DOS;
        PrvProgram = Program;
        PrvService = Service;
        PrvStaffID = StaffID;
        PreviousClient = currentClient;
        PrvEpisode = (int)row["EPISODE"];
    }

}

private void Test_Test_Load(object sender, EventArgs e)
{
    this.Text += " Rule: " + avConn.CurrentRuleTitle + " Type: " +
        avConn.currentServiceType.ToString() + " Billable Code: " +
        avConn.BillableCode;
    this.comboBox1.DataSource = Enum.GetValues(typeof(Avatar_Solutions.month));

    comboBox2.Text = DateTime.Today.ToString("yyyy");
    comboBox1.SelectedIndex = DateTime.Today.Month - 2;
}

private void button1_Click(object sender, EventArgs e)

```

```
264     {
265         dtbl.Clear();
266         this.GetData();
267         comboBox1.SelectedIndex += 1;
268     }
269
270 private void button2_Click(object sender, EventArgs e)
271 {
272     this.ProcessData();
273
274     Report_Viewer frm = new
275     Report_Viewer(rptList, TotalCreated, TotalAlreadyCreated, TotalError, avConn.CurrentRuleTitle, From.Value, ThruDate.Value);
276     frm.ShowDialog();
277
278     // SendMessage();
279 }
280
281 }
282 }
283 }
284 }
```