

```

// Easily created and modified report programs
// Ted Holt
// QIWS must be in your library list in order to compile.

1 H dftactgrp(*no) actgrp(*caller) option(*srcstmt: *nodebugio)
2
3 Fqcustcdt  if  e          disk  prefix(C_)
4 Fqsysprt  o  f 132      printer oflind(Overflow)
5 F          prtctl(PrtCtl)
6
7 * printer control data structure; controls spacing and skipping
8
9 D PrtCtl          ds
10 D SpaceAndSkip      1      12
11 D SpaceBefore       1       3
12 D SpaceAfter        4       6
13 D SkipBefore        7       9
14 D SkipAfter         10      12
15 D CurrentLine       13     15s 0
16
17 * this save field is required for double-strike
18
19 D SaveSpaceAndSkip...
20 D          s          like(SpaceAndSkip)
21
22 * predefined constants for spacing and skipping
23
24 D SingleSpace      c          '001'
25 D DoubleSpace     c          '002'
26 D NoSpacing       c          '000'
27 D TopOfForm       c          '001'
28
29 * print control variables
30
31 D DoubleStrike     s          n
32 D eof              s          n
33 D Overflow         s          n
34 D PageNbr         s          3p 0
35
36 * report definition
37
38 D PrintLine        ds          132
39 D Column01         7
40 D                  1
41 D Column02         25
42 D                  1
43 D Column03         7
44 D                  2
45 D Column04         15
46 D                  3
47 D Column05         12
48 D                  1
49 D Column06         12
50 D                  1
51 D Column07         2
52
53 * system definitions
54
55 D psds            sds          qualified
56 D ProcName        *proc
57 D RunDate         6s 0 overlay(psds:276)
58 D RunTime         6s 0 overlay(psds:282)
59
60 D True            c          const(*on)
61 D False          c          const(*off)
62

```

```
59 * control break fields (comparison fields and accumulators)
60
61 D SaveState      s           like(C_State)
62 D CdtDuel       s           like(C_CdtDue)
63 D CdtDuelR      s           like(CdtDuel)
64 D BalDuel       s           like(C_BalDue)
65 D BalDuelR      s           like(BalDuel)
66
67 /free
68     *inlr = *on;
69
70     exsr PrintHeaders;
71     exsr Read;
72
73     dow not eof;
74         exsr PrintOneState;
75     enddo;
76
77     // print the report totals
78
79     eval Column02 = '    Grand total';
80     evalr Column05 = %editc(CdtDuelR:'J':*CurSym);
81     evalr Column06 = %editc(BalDuelR:'J':*CurSym);
82     eval Column07 = '***';
83     eval SpaceBefore = SingleSpace;
84     exsr Print;
85
86     eval SpaceBefore = DoubleSpace;
87     eval PrintLine = '** End of report **';
88     exsr Print;
89     return;
90
91     // =====
92     // Read one input record
93     // =====
94     begsr Read;
95
96         read cusrec;
97         eof = %eof();
98
99     endsr;
100    // =====
101    // Process all the records for a state
102    // =====
103    begsr PrintOneState;
104
105        // save the state for comparison to detect the control break
106        SaveState = C_State;
107
108        // clear the accumulators for the state
109        CdtDuel = *zero;
110        BalDuel = *zero;
111
112        dow not eof
113        and C_State = SaveState;
114
115            // start a new page if necessary
116            if Overflow;
117                exsr PrintHeaders;
118            endif;
119
120            exsr PrintDetail;
121            // accumulate the money fields
122            CdtDuel += C_CdtDue;
123            BalDuel += C_BalDue;
124            exsr Read;
```

```

125
126     enddo;
127
128     // print the state totals
129     evalr      Column05 = *all'-';
130     evalr      Column06 = *all'-';
131     exsr      Print;
132
133     eval      Column02 = '      ' + SaveState + ' total';
134     evalr     Column05 = %editc(CdtDuel:'J':*CurSym);
135     evalr     Column06 = %editc(BalDuel:'J':*CurSym);
136     eval      Column07 = '*';
137     eval      SpaceBefore = SingleSpace;
138     eval      SpaceAfter  = SingleSpace;
139     exsr      Print;
140
141     // roll state totals into grand totals
142     CdtDueR += CdtDuel;
143     BalDueR += BalDuel;
144
145 endsr;
146 // =====
147 // Process one input record
148 // =====
149 begsr PrintDetail;
150
151     eval      Column01 = C_State;
152     eval      Column02 = %trimr(C_LstNam) + ' ' +
153                       C_Init;
154     evalr     Column03 = %editc(C_CusNum:'4');
155     eval      Column04 = C_City;
156     evalr     Column05 = %editc(C_CdtDue:'J');
157     evalr     Column06 = %editc(C_BalDue:'J');
158     exsr      Print;
159
160 endsr;
161 // =====
162 // Print page and column headings
163 // =====
164 begsr PrintHeaders;
165
166     // print the headers in bold face
167     DoubleStrike = True;
168
169     PageNbr += 1;
170     PrintLine = 'Customer List by State (' +
171               %trim(psdS.ProcName) + ')      ' +
172               %editc(psdS.RunDate:'Y') + ' ' +
173               %editw(psdS.RunTime:' : : ') +
174               ' Page ' + %char(PageNbr);
175     SkipBefore = TopOfForm;
176     exsr Print;
177
178     evalr     Column03 = 'Account';
179     evalr     Column05 = 'Credit';
180     evalr     Column06 = 'Balance';
181     eval      SpaceBefore = DoubleSpace;
182     exsr      Print;
183
184     eval      Column01 = 'State';
185     eval      Column02 = 'Name';
186     evalr     Column03 = 'Number';
187     eval      Column04 = 'City';
188     evalr     Column05 = 'Due';
189     evalr     Column06 = 'Due';
190     eval      SpaceBefore = SingleSpace;

```

```
191         exsr         Print;
192
193         eval         Column01 = *all'=';
194         eval         Column02 = *all'=';
195         eval         Column03 = *all'=';
196         eval         Column04 = *all'=';
197         eval         Column05 = *all'=';
198         eval         Column06 = *all'=';
199         eval         SpaceBefore = SingleSpace;
200         exsr         Print;
201
202         eval         DoubleStrike = False;
203         eval         Overflow = *off;
204
205     endsr;
206     // =====
207     // Write to the printer
208     // =====
209     begsr Print;
210
211         if SpaceAndSkip = *blanks;
212             SpaceBefore = SingleSpace;
213         endif;
214
215         if DoubleStrike;
216             SaveSpaceAndSkip = SpaceAndSkip;
217             SkipAfter = NoSpacing;
218             SpaceAfter = NoSpacing;
219             write Qsysprt PrintLine;
220             SpaceAndSkip = SaveSpaceAndSkip;
221             SkipBefore = NoSpacing;
222             SpaceBefore = NoSpacing;
223         endif;
224         write Qsysprt PrintLine;
225
226         reset PrintLine;
227         reset SpaceAndSkip;
228
229     endsr;
```